

ΚΕΦΑΛΑΙΟ 10^ο

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

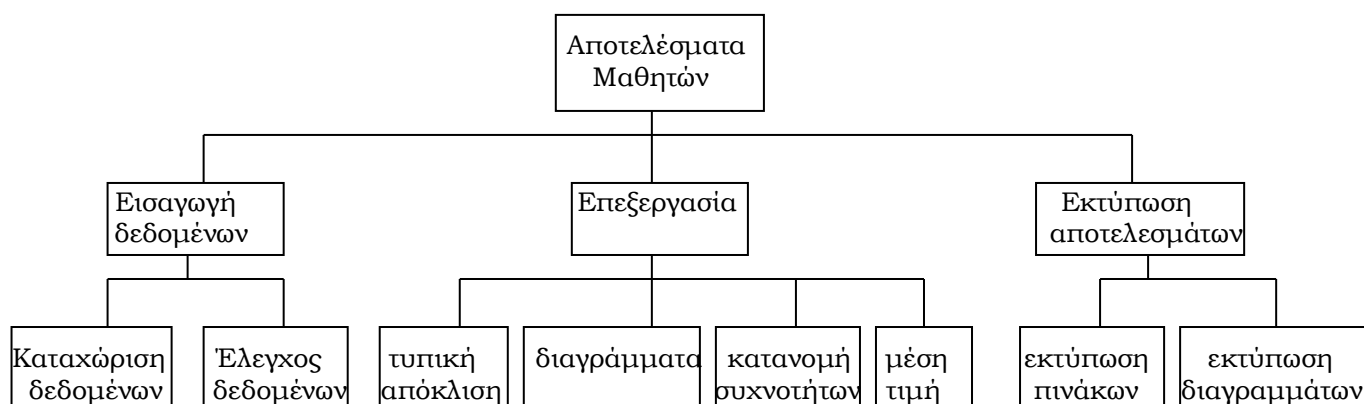
ΘΕΩΡΙΑ

10.1 Τμηματικός προγραμματισμός

Όπως έχει αναφερθεί, ο καλύτερος τρόπος αντιμετώπισης σύνθετων προβλημάτων είναι η ιεραρχική προσέγγιση, δηλαδή η διάσπαση του προβλήματος σε μικρότερα επιμέρους προβλήματα και καθενός από αυτά σε άλλα, ακόμα απλούστερα και μικρότερα, κ.ο.κ. Με αυτό το τρόπο ο τελικός αλγόριθμος του προβλήματος ανάγεται σε πολλούς απλούστερους επί μέρους αλγόριθμους και το τελικό πρόγραμμα σε πολλά απλούστερα τμήματα προγράμματος. Η τεχνική αυτή του τμηματικού προγραμματισμού αποτελεί βασικό συστατικό του δομημένου προγραμματισμού και εξασφαλίζει την επιτυχή και εύκολη δημιουργία σωστών προγραμμάτων.

Ορισμός: Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

Παράδειγμα: Έστω ότι θέλουμε να αντιμετωπίσουμε το πρόβλημα του 1^{ου} κεφαλαίου, δηλαδή την αξιολόγηση των αποτελεσμάτων των μαθητών της Γ' Λυκείου στα μαθήματα ειδικότητας. Εφαρμόζοντας την ανάλυση του προβλήματος σε άλλα απλούστερα φτάνουμε στην εξής διαγραμματική αναπαράσταση:



Υποπρόγραμμα: Όταν ένα τμήμα προγράμματος επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα, τότε χαρακτηρίζεται ως υποπρόγραμμα.

10.2 Χαρακτηριστικά των υποπρογραμμάτων

Η διάσπαση του αρχικού προβλήματος σε μικρότερα υποπροβλήματα οδηγεί σε μικρότερα τμήματα προγραμμάτων που ονομάζονται υποπρογράμματα. Η ανάλυση αυτή δεν είναι πάντα εύκολη και δεν υπάρχει συγκεκριμένη μεθοδολογία. Απαιτεί μελέτη στην ανάλυση του προβλήματος, εμπειρία, ταλέντο και γνώσεις. Υπάρχουν 3 ιδιότητες που πρέπει να διακρίνουν τα υποπρογράμματα:

- **Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο.** Το υποπρόγραμμα ενεργοποιείται με την είσοδό του που γίνεται στην αρχή του, εκτελεί ορισμένες ενέργειες και απενεργοποιείται από την έξοδό του που γίνεται στο τέλος του.
- **Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα.** Η σχεδίαση, ανάπτυξη και συντήρηση κάθε υποπρογράμματος μπορεί να γίνει αυτόνομα, χωρίς να επηρεαστούν τα υπόλοιπα. Στην πράξη η απόλυτη ανεξαρτησία είναι δύσκολο να επιτευχθεί.
- **Κάθε υποπρόγραμμα πρέπει να μην είναι μεγάλο.** Πρέπει να εκτελεί μόνο μία λειτουργία. Αν εκτελεί περισσότερες, πρέπει να διασπαστεί σε μικρότερα υποπρογράμματα.

10.3 Πλεονεκτήματα του τμηματικού προγραμματισμού

Όπως αναφέρθηκε συνοπτικά στο κεφάλαιο 6, ο τμηματικός προγραμματισμός παρουσιάζει πολλά πλεονεκτήματα και μας οδηγεί στη σύνθεση σωστών προγραμμάτων.

- **Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντίστοιχου προγράμματος.** Αυτό, γιατί εξετάζουμε και επιλύουμε το επιμέρους απλό υποπρόβλημα και όχι ολικά το αρχικό πρόβλημα.
- **Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.** Κάθε αυτοτελές τμήμα προγράμματος, λόγω της απλότητας και της έκτασής του είναι εύκολο να διορθωθεί και να κατανοηθεί από τους προγραμματιστές.
- **Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.** Πολλές φορές χρειάζεται η ίδια λειτουργία σε διαφορετικά σημεία του προγράμματος. Η ύπαρξη υποπρογράμματος για αυτή τη λειτουργία και η κλήση του στα σημεία του προγράμματος που χρειάζεται, μειώνει το μέγεθος του συνολικού προγράμματος, τις πιθανότητες λάθους και το χρόνο συγγραφής.

- **Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.** Μία λειτουργία που μπορεί να την χρειαζόμαστε συχνά στα προγράμματά μας, όπως π.χ. η εύρεση του μικρότερου δύο αριθμών, μπορεί να γίνεται με την κλήση ενός υποπρογράμματος που έχουμε γράψει γι'αυτή τη λειτουργία. Η χρήση τέτοιων υποπρογραμμάτων είναι σαν τη χρήση των ενσωματωμένων συναρτήσεων που διαθέτει η ΓΛΩΣΣΑ.

10.4 Παράμετροι

Κάθε υποπρόγραμμα, για να ενεργοποιηθεί και να εκτελέσει τις λειτουργίες του, καλείται από το αρχικό πρόγραμμα ή ένα άλλο υποπρόγραμμα που ονομάζεται κυρίως πρόγραμμα. Δέχεται λοιπόν τιμές από το κυρίως πρόγραμμα και μετά την εκτέλεσή του επιστρέφει στο κυρίως πρόγραμμα κάποια αποτελέσματα. Για το πέρασμα τιμών σε ένα υποπρόγραμμα χρησιμοποιούνται ειδικές μεταβλητές που ονομάζονται παράμετροι.

Μία παράμετρος είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.

10.5 Διαδικασίες και συναρτήσεις

Υπάρχουν δύο ειδών υποπρογράμματα, οι διαδικασίες και οι συναρτήσεις.

Οι διαδικασίες μπορούν να εκτελέσουν όλες τις λειτουργίες που μπορεί να εκτελέσει ένα πρόγραμμα, δηλαδή:

- να εισάγουν δεδομένα
- να εκτελέσουν υπολογισμούς
- να μεταβάλλουν τις τιμές των μεταβλητών
- να τυπώσουν αποτελέσματα

Επίσης, με τη χρήση των παραμέτρων μπορούν να μεταφέρουν τις τιμές των μεταβλητών τους στο κυρίως πρόγραμμα ή σε άλλα υποπρογράμματα.

Οι συναρτήσεις έχουν πιο περιορισμένη λειτουργία, αφού υπολογίζουν μόνο μία τιμή, αριθμητική, χαρακτήρα ή λογική και αυτήν επιστρέφουν στο κυρίως πρόγραμμα. Η χρήση τους είναι όμοια με τη χρήση των ενσωματωμένων συναρτήσεων της ΓΛΩΣΣΑΣ.

Τόσο οι διαδικασίες όσο και οι συναρτήσεις τοποθετούνται μετά το τέλος του κυρίως προγράμματος.

10.5.1 Ορισμός και κλήση συναρτήσεων

Η γενική σύνταξη κάθε συνάρτησης έχει ως εξής:

ΣΥΝΑΡΤΗΣΗ όνομα_συνάρτησης (λίστα παραμέτρων): τύπος συνάρτησης

Τμήμα δηλώσεων

ΑΡΧΗ

....

όνομα_συνάρτησης←έκφραση

....

ΤΕΛΟΣ ΣΥΝΑΡΤΗΣΗΣ

- **όνομα_συνάρτησης:** Είναι το όνομα της συνάρτησης και ακολουθεί τους κανόνες ονοματολογίας των μεταβλητών.
- **λίστα παραμέτρων:** Είναι μεταβλητές χωρισμένες με κόμμα και οι τιμές τους μεταβιβάζονται στη συνάρτηση κατά την κλήση της από το κυρίως πρόγραμμα.
- **τύπος συνάρτησης:** Μπορεί να είναι ΑΚΕΡΑΙΟΣ, ΠΡΑΓΜΑΤΙΚΟΣ, ΧΑΡΑΚΤΗΡΑΣ, ΛΟΓΙΚΟΣ. Περιγράφει τον τύπο της τιμής που η συνάρτηση επιστρέφει στο κυρίως πρόγραμμα.
- **Τμήμα δηλώσεων:** Δηλώνουμε τις μεταβλητές και συμβολικές σταθερές που χρησιμοποιούμε στη συνάρτηση, με τον ίδιο τρόπο όπως και σε ένα πρόγραμμα. Στο τμήμα δηλώσεων πρέπει να δηλώσουμε και τις μεταβλητές της λίστας παραμέτρων.
- **όνομα_συνάρτησης←έκφραση:** Πρέπει οπωσδήποτε να είναι μία από τις εντολές που χρησιμοποιούμε στη συνάρτηση, π.χ. αν έχουμε φτιάξει μία συνάρτηση με όνομα ΜΟ που δέχεται δύο αριθμούς α,β σαν παραμέτρους και υπολογίζει το μέσο όρο τους, η εντολή εκχώρησης θα ήταν: ΜΟ←(α+β)/2.

Σημείωση: Απαγορεύεται μέσα στις εντολές μίας συνάρτησης να έχουμε εντολές εισόδου/εξόδου δεδομένων. Δηλαδή απαγορεύεται η χρήση των εντολών ΔΙΑΒΑΣΕ, ΓΡΑΨΕ, ΕΜΦΑΝΙΣΕ, ΤΥΠΩΣΕ, ΕΚΤΥΠΩΣΕ. Αν σε ένα υποπρόγραμμα υπάρχει ανάγκη για είσοδο ή έξοδο δεδομένων, τότε πρέπει να χρησιμοποιήσουμε διαδικασία και όχι συνάρτηση.

Παράδειγμα

Έστω ότι θέλουμε να φτιάξουμε μία συνάρτηση με όνομα maximum που θα δέχεται δύο ακέραιους και θα επιστρέφει το μεγαλύτερο από αυτούς:

```
ΣΥΝΑΡΤΗΣΗ maximum(α,β):ΑΚΕΡΑΙΑ  
ΜΕΤΑΒΛΗΤΕΣ  
ΑΚΕΡΑΙΕΣ:α,β  
ΑΡΧΗ  
  ΑΝ α>β ΤΟΤΕ  
    maximum ← α  
  ΑΛΛΙΩΣ  
    maximum ← β  
  ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Στο κυρίως πρόγραμμα, αν έχουμε διαβάσει δύο μεταβλητές X,Y και θέλουμε με τη χρήση της συνάρτησης να εμφανίσουμε το μεγαλύτερο, θα γράφαμε:

```
max←maximum(X,Y)           ή           ΓΡΑΨΕ max  
ΓΡΑΨΕ maximum(X,Y)
```

Ο μηχανισμός κλήσης της συνάρτησης έχει ως εξής: Στο κυρίως πρόγραμμα οι τιμές των μεταβλητών X και Y είναι γνωστές (έχουν διαβαστεί ή έχουν εκχωρηθεί τιμές σε αυτές). Κατά την κλήση, διακόπτεται η εκτέλεση του κυρίως προγράμματος και οι τιμές των X και Y μεταβιβάζονται αντίστοιχα στις μεταβλητές α και β της συνάρτησης. Η συνάρτηση υπολογίζει το μεγαλύτερο των α και β, δηλαδή στην ουσία των X και Y και εκχωρείται στο όνομα της συνάρτησης ο μεγαλύτερος. Μετά το τέλος της συνάρτησης γίνεται επιστροφή στο κυρίως πρόγραμμα.

10.5.2 Ορισμός και κλήση διαδικασιών

Η γενική σύνταξη κάθε διαδικασίας έχει ως εξής:

```
ΔΙΑΔΙΚΑΣΙΑ όνομα_διαδικασίας (λίστα παραμέτρων)  
Τμήμα δηλώσεων  
ΑΡΧΗ  
  εντολές  
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

- **όνομα_διαδικασίας:** Είναι το όνομα της διαδικασίας και ακολουθεί τους κανόνες ονοματολογίας των μεταβλητών.
- **λίστα παραμέτρων:** Είναι μεταβλητές χωρισμένες με κόμμα και οι τιμές τους μεταβιβάζονται στη διαδικασία κατά την κλήση της από το κυρίως πρόγραμμα και επιστρέφουν τις τιμές τους (σε αντίθεση με τις συναρτήσεις) στο κυρίως πρόγραμμα. Στις διαδικασίες, σε αντίθεση με τις συναρτήσεις, η λίστα παραμέτρων δεν είναι υποχρεωτική π.χ. μπορεί μία διαδικασία να τυπώνει απλά ένα μήνυμα στην οθόνη.
- **Τμήμα δηλώσεων:** Δηλώνουμε τις μεταβλητές και συμβολικές σταθερές που χρησιμοποιούμε στη διαδικασία, με τον ίδιο τρόπο όπως και σε ένα πρόγραμμα. Στο τμήμα δηλώσεων πρέπει να δηλώσουμε και τις μεταβλητές της λίστας παραμέτρων.
- **εντολές:** οποιοσδήποτε εντολές της ΓΛΩΣΣΑΣ.

Κάθε διαδικασία εκτελείται όταν καλείται από το κυρίως πρόγραμμα ή άλλη διαδικασία. Η κλήση της διαδικασίας πραγματοποιείται με την εντολή **ΚΑΛΕΣΕ** ακολουθούμενο από το όνομα της διαδικασίας και της λίστας παραμέτρων μέσα σε παρενθέσεις, π.χ. **ΚΑΛΕΣΕ** Πράξεις(A, B , Διαφορά).

Παράδειγμα

Έστω ότι θέλουμε να φτιάξουμε μία διαδικασία με όνομα minimum που θα δέχεται δύο ακέραιους και θα επιστρέφει το μικρότερο από αυτούς:

```
ΔΙΑΔΙΚΑΣΙΑ minimum (α, β, γ)  
ΜΕΤΑΒΛΗΤΕΣ  
ΑΚΕΡΑΙΕΣ:α,β,γ  
ΑΡΧΗ  
  ΑΝ α<β ΤΟΤΕ  
    γ← α  
  ΑΛΛΙΩΣ  
    γ← β  
  ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Στο κυρίως πρόγραμμα, αν έχουμε για παράδειγμα τις μεταβλητές κ,λ και θέλουμε να εμφανίσουμε τον μικρότερο των δύο θα γράφαμε τις εντολές:

ΚΑΛΕΣΕ minimum (κ, λ, μ)

ΓΡΑΨΕ μ

Ο μηχανισμός κλήσης της διαδικασίας έχει ως εξής: Κατά την κλήση της διαδικασίας, διακόπτεται η εκτέλεση του κυρίως προγράμματος και μεταβιβάζονται οι τιμές των κ,λ,μ στις μεταβλητές α,β,γ αντίστοιχα της διαδικασίας. Στη συνέχεια, εκτελούνται οι εντολές της διαδικασίας και υπολογίζεται ο μικρότερος στη μεταβλητή γ. Όταν τελειώνει η διαδικασία επιστρέφονται οι τιμές των α,β,γ της διαδικασίας στις κ,λ,μ κατά αντιστοιχία του κυρίως προγράμματος. Έτσι, η μεταβλητή μ περιέχει τον μικρότερο των κ, λ. Στη συνέχεια εκτελείται η αμέσως επόμενη εντολή μετά την ΚΑΛΕΣΕ.

Παράδειγμα χρήσης διαδικασίας και συνάρτησης (σχολικού βιβλίου)

Να γραφεί πρόγραμμα που να υπολογίζει το εμβαδό του κύκλου από την ακτίνα του.

Λύση

Το πρόγραμμα θα πρέπει να κάνει τα εξής:

1. Διαβάζει την ακτίνα του κύκλου που πρέπει να είναι θετικός αριθμός.
2. Υπολογίζει το εμβαδό του κύκλου ($E=\pi r^2$).
3. Εμφανίζει το εμβαδό.

Αν και είναι πολύ απλή η λύση και χωρίς τη χρήση υποπρογραμμάτων, ας το διασπάσουμε σε τρία υποπρογράμματα, ένα για κάθε λειτουργία:

1. Το πρώτο υποπρόγραμμα πρέπει να διαβάζει την ακτίνα του κύκλου και να την επιστρέφει στο κυρίως πρόγραμμα. Θα φτιάξουμε λοιπόν μία διαδικασία (δεν μπορούμε να χρησιμοποιήσουμε συνάρτηση γιατί γίνεται εισαγωγή δεδομένων) με όνομα Είσοδος_δεδομένων που θα δέχεται από το πληκτρολόγιο την τιμή της ακτίνας (πρέπει να είναι θετικός αριθμός) και θα την καταχωρεί στη μεταβλητή Αριθμός:

ΔΙΑΔΙΚΑΣΙΑ Είσοδος_δεδομένων (Αριθμός)

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ : Αριθμός

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε την ακτίνα'

ΔΙΑΒΑΣΕ Αριθμός

ΜΕΧΡΙΣ_ΟΤΟΥ Αριθμός > 0

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

2. Το δεύτερο υποπρόγραμμα θα υπολογίζει το εμβαδό του κύκλου και θα επιστρέφει τη τιμή στο κυρίως πρόγραμμα. Εφόσον θα επιστρέφει μία τιμή, θα το υλοποιήσουμε με συνάρτηση (μπορούμε επίσης και με διαδικασία). Θα φτιάξουμε τη συνάρτηση Εμβαδό_κύκλου που θα δέχεται την τιμή της ακτίνας του κύκλου και θα επιστρέφει με το όνομά της το εμβαδό:

ΣΥΝΑΡΤΗΣΗ Εμβαδό_κύκλου (ακτίνα) : **ΠΡΑΓΜΑΤΙΚΗ**

ΣΤΑΘΕΡΕΣ

$\pi=3.14$

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ακτίνα

ΑΡΧΗ

Εμβαδό_κύκλου <- $\pi * \text{ακτίνα}^2$

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

3. Το τρίτο υποπρόγραμμα θα εκτυπώνει το αποτέλεσμα, δηλαδή το εμβαδό του κύκλου. Θα φτιάξουμε τη διαδικασία Εκτύπωση η οποία θα δέχεται από το κυρίως πρόγραμμα μία τιμή και θα την εκτυπώνει:

ΔΙΑΔΙΚΑΣΙΑ Εκτύπωση (Αποτέλεσμα)

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ : Αποτέλεσμα

ΑΡΧΗ

ΓΡΑΨΕ 'Το εμβαδό του κύκλου είναι :', Αποτέλεσμα

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Το κυρίως πρόγραμμα που θα κάνει χρήση των τριών αυτών υποπρογραμμάτων θα είναι:

ΠΡΟΓΡΑΜΜΑ Υπολογισμός_Εμβαδού
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ : R, E
ΑΡΧΗ
ΚΑΛΕΣΕ Είσοδος_δεδομένων(R)
E ← Εμβαδό_κύκλου(R)
ΚΑΛΕΣΕ Εκτύπωση(E)
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΕΚΤΕΛΕΣΗ:

1. Με την εντολή **ΚΑΛΕΣΕ** Είσοδος_δεδομένων(R) σταματά η εκτέλεση του κυρίως προγράμματος και η τιμή της μεταβλητής R μεταβιβάζεται στη μεταβλητή Αριθμός της διαδικασίας. Μετά το πέρας της διαδικασίας Είσοδος_δεδομένων έχει διαβαστεί η μεταβλητή Αριθμός και κατά την επιστροφή στο κυρίως πρόγραμμα μεταβιβάζεται η τιμή της στη μεταβλητή R του κυρίως προγράμματος. Άρα, πλέον η μεταβλητή R του κυρίως προγράμματος περιέχει τιμή (αυτή που διαβάστηκε μέσα στη διαδικασία). Μετά εκτελείται η επόμενη εντολή.
2. Στη συνέχεια, με την εντολή **E ← Εμβαδό_κύκλου(R)**, η τιμή της R μεταβιβάζεται στη μεταβλητή ακτίνα της συνάρτησης και η συνάρτηση επιστρέφει το εμβαδό του κύκλου με το όνομά της. Έτσι, η μεταβλητή E του κυρίως προγράμματος θα περιέχει το εμβαδό.
3. Με την εντολή **ΚΑΛΕΣΕ** Εκτύπωση(E), η τιμή της μεταβλητής E, δηλ. το εμβαδό του κύκλου, μεταβιβάζεται στη μεταβλητή Αποτέλεσμα της διαδικασίας, και μέσα στη διαδικασία εκτυπώνεται η τιμή της στην οθόνη.

10.5.3 Πραγματικές και τυπικές παράμετροι

Για την καλύτερη κατανόηση ανταλλαγής τιμών ανάμεσα στις παραμέτρους κατά την κλήση μιας διαδικασίας και κατά την επιστροφή από αυτήν, ας θεωρήσουμε το ακόλουθο κυρίως πρόγραμμα και διαδικασία:

ΠΡΟΓΡΑΜΜΑ Παράδειγμα

....
A ← 5
B ← 7
ΚΑΛΕΣΕ Πράξεις(A, B, Διαφ1, Αθρ1)
....

ΔΙΑΔΙΚΑΣΙΑ Πράξεις (X, Y, Διαφορά, Άθροισμα)

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: X, Y, Διαφορά, Άθροισμα

ΑΡΧΗ

Διαφορά ← X - Y

Άθροισμα ← X + Y

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Ας δούμε καλύτερα το μηχανισμό κλήσης και επιστροφής της διαδικασίας:

1. **ΚΑΛΕΣΕ** Πράξεις(A, B, Διαφ1, Αθρ1): Οι τιμές των μεταβλητών A, B, Διαφ1, Αθρ1 του κυρίως προγράμματος μεταβιβάζονται κατά αντιστοιχία στις μεταβλητές X, Y, Διαφορά, Άθροισμα, της διαδικασίας:

A	5	→	5	X
B	7	→	7	Y
Διαφ1		→		Διαφορά
Αθρ1		→		Άθροισμα

2. **Εκτέλεση της διαδικασίας:** Η μεταβλητή Διαφορά της διαδικασίας γίνεται -2 και η μεταβλητή Άθροισμα γίνεται 12:

5	X
7	Y
-2	Διαφορά
12	Άθροισμα

3. **Επιστροφή στο κυρίως πρόγραμμα:** Μετά το **ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ** οι τιμές των X, Y, Διαφορά, Άθροισμα της διαδικασίας μεταβιβάζονται κατά αντιστοιχία στις μεταβλητές A, B, Διαφ1, Αθρ1 του κυρίως προγράμματος:

A	5	←	5	X
B	7	←	7	Y
Διαφ1	-2	←	-2	Διαφορά
Αθρ1	12	←	12	Άθροισμα

Στη συνέχεια εκτελείται η αμέσως επόμενη εντολή μετά την **ΚΑΛΕΣΕ** Πράξεις(A, B, Διαφ1, Αθρ1).

Πραγματικές παράμετροι: είναι οι παράμετροι κατά την κλήση του υποπρογράμματος.

Τυπικές παράμετροι: είναι οι παράμετροι στη δήλωση του υποπρογράμματος.

Στο παράδειγμά μας οι μεταβλητές A, B, Διαφ1, Αθρ1 του κυρίως προγράμματος είναι οι πραγματικές παράμετροι, ενώ οι μεταβλητές X, Y, Διαφορά, Αθροισμα της διαδικασίας είναι οι τυπικές παράμετροι.

Οι μεταβλητές του κυρίως προγράμματος είναι άγνωστες στη διαδικασία και αντίστοιχα οι μεταβλητές που χρησιμοποιούνται στη διαδικασία είναι άγνωστες στο κυρίως πρόγραμμα. Π.χ. η διαδικασία Πράξεις δεν γνωρίζει κάποια μεταβλητή με όνομα A και ομοίως το κυρίως πρόγραμμα δεν γνωρίζει κάποια μεταβλητή με όνομα X. Οι πραγματικές παράμετροι είναι γνωστές μόνο στο κυρίως πρόγραμμα και οι τυπικές παράμετροι είναι γνωστές μόνο στη διαδικασία, δηλαδή έχουν όπως λέμε **τοπική ισχύ**. Πρόκειται δηλαδή για διαφορετικές θέσεις στη μνήμη του υπολογιστή. Γι'αυτό το λόγο οι πραγματικές και τυπικές παράμετροι μπορούν να έχουν το ίδιο όνομα.

Οι πραγματικές και τυπικές παράμετροι πρέπει να ακολουθούν τους εξής κανόνες:

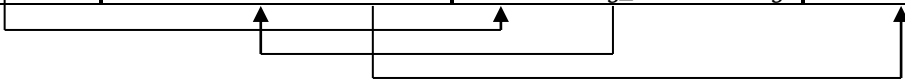
1. Ο αριθμός πραγματικών και τυπικών παραμέτρων πρέπει να είναι ο ίδιος.
2. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση.
3. Η πραγματική παράμετρος και η αντίστοιχή της τυπική πρέπει να είναι του ίδιου τύπου.

Η χρήση στοίβας στη κλήση διαδικασιών

Η έννοια της στοίβας είναι πολύ χρήσιμη στο ίδιο το λογισμικό των γλωσσών προγραμματισμού. Όταν μια διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κυρίου προγράμματος, που ονομάζεται διεύθυνση επιστροφής (return address), αποθηκεύεται από το μεταφραστή σε μια στοίβα που ονομάζεται στοίβα χρόνου εκτέλεσης (execution time stack). Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης, η διεύθυνση επιστροφής απωθείται από τη στοίβα και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κύριο πρόγραμμα. Η τεχνική αυτή εφαρμόζεται και γενικότερα, δηλαδή οποτεδήποτε μια διαδικασία ή συνάρτηση καλεί μια άλλη διαδικασία ή συνάρτηση.

Ας δούμε καλύτερα τη λειτουργία της στοίβας χρόνου εκτέλεσης με τη βοήθεια του παρακάτω παραδείγματος που περιλαμβάνει το κύριο πρόγραμμα και τρεις διαδικασίες:

ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ	ΔΙΑΔΙΚΑΣΙΑ Δ1	ΔΙΑΔΙΚΑΣΙΑ Δ2	ΔΙΑΔΙΚΑΣΙΑ Δ3
385 εντολή 1	510 εντολή 1	652 εντολή 1	890 εντολή 1
386 εντολή 2	511 ΚΑΛΕΣΕ Δ3	653 ΚΑΛΕΣΕ Δ1	891 εντολή 2
387 ΚΑΛΕΣΕ Δ2	512 εντολή 3	654 εντολή 3	892 εντολή 3
388 εντολή 4	513 Τέλος_Διαδικασίας	655 εντολή 4	893 Τέλος_Διαδικασίας
390		656 Τέλος_Διαδικασίας	



Οι αριθμοί αριστερά της κάθε εντολής είναι η διεύθυνση στη μνήμη του υπολογιστή που αυτή είναι αποθηκευμένη. Η διαμόρφωση της στοίβας χρόνου εκτέλεσης θα έχει ως εξής:

Εκτέλεση εντολών	Στοιβα χρόνου εκτέλεσης
Εκτελούνται οι εντολή 1 και εντολή 2 του κυρίως προγράμματος. Η στοίβα αρχικά είναι άδεια.	5 4 3 2 1 top=0
Καλείται η διαδικασία Δ2 και η διεύθυνση της επόμενης εντολής (388) ωθείται στη στοίβα. Ο έλεγχος πηγαίνει στη διαδικασία Δ2.	5 4 3 2 1 top=1 388
Εκτελείται η εντολή 1 της διαδικασίας Δ2 και στη συνέχεια καλείται η διαδικασία Δ1, οπότε ωθείται στη στοίβα η διεύθυνση της επόμενης εντολής δηλ. η 654.	5 4 3 2 1 top=2 654 388

Εκτελείται η εντολή 1 της διαδικασίας Δ1 και στη συνέχεια καλείται η διαδικασία Δ3, οπότε ωθείται στη στοίβα η διεύθυνση της επόμενης εντολής δηλ. η 512 .	5 4 top=3 2 1 512 654 388
Εκτελούνται οι τρεις εντολές της διαδικασίας Δ3 και στην εντολή Τέλος_Διαδικασίας απωθείται από τη στοίβα η διεύθυνση 512, δηλαδή η διεύθυνση της εντολής που θα εκτελεστεί στη συνέχεια.	5 4 3 top=2 1 654 388
Εκτελείται η εντολή 3 της διαδικασίας Δ1 και με το τέλος της διαδικασίας απωθείται από τη στοίβα η διεύθυνση επιστροφής 654.	5 4 3 2 top=1 388
Εκτελούνται οι εντολές 3 και 4 της διαδικασίας Δ2 και με το τέλος της διαδικασίας απωθείται από τη στοίβα η διεύθυνση 388 που είναι η διεύθυνση της εντολής 4 του κυρίως προγράμματος.	5 4 3 2 1 top=0

Στη συνέχεια εκτελείται η εντολή 4 του κυρίως προγράμματος.

10.6 Εμβέλεια μεταβλητών-σταθερών.

Κάθε κύριο πρόγραμμα όπως και κάθε υποπρόγραμμα περιλαμβάνει τις δικές του μεταβλητές και σταθερές. Οι μεταβλητές αυτές στη **ΓΛΩΣΣΑ** είναι γνωστές στο αντίστοιχο υποπρόγραμμα και μόνο σε αυτό. Όλες οι μεταβλητές (και οι συμβολικές σταθερές) είναι τοπικές στο συγκεκριμένο τμήμα προγράμματος.

Ο μόνος τρόπος για να περάσει μία τιμή από ένα υποπρόγραμμα σε ένα άλλο ή από το κυρίως πρόγραμμα σε ένα υποπρόγραμμα είναι διαμέσου των παραμέτρων κατά το στάδιο της κλήσης του υποπρογράμματος και μετά το τέλος της εκτέλεσης του υποπρογράμματος.

Ας δούμε το παρακάτω παράδειγμα:

ΠΡΟΓΡΑΜΜΑ Αρχικό ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: Α , Β , Γ ΑΡΧΗ ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ	ΔΙΑΔΙΚΑΣΙΑ Πρώτη ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: Δ , Ε , Ζ , Η ΑΡΧΗ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	ΔΙΑΔΙΚΑΣΙΑ Δεύτερη ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: Γ ,Θ , Ι ΑΡΧΗ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
--	---	---

Οι μεταβλητές του προγράμματος Αρχικό με ονόματα Α,Β,Γ είναι γνωστές, δηλ. ισχύουν μόνο για το πρόγραμμα. Έξω από το πρόγραμμα, σε όλα τα υποπρογράμματα οι μεταβλητές αυτές δεν ισχύουν. Επίσης, η διαδικασία Πρώτη έχει τις πραγματικές μεταβλητές Δ,Ε,Ζ,Η, οι οποίες ισχύουν μόνο για τη συγκεκριμένη διαδικασία και όχι για τα υπόλοιπα υποπρογράμματα ή το κύριο πρόγραμμα.

Η διαδικασία Δεύτερη έχει και αυτή τις δικές της μεταβλητές Γ,Θ,Ι. Οι μεταβλητές αυτές ισχύουν μόνο για τη διαδικασία Δεύτερη. Η μεταβλητή με όνομα Γ δεν έχει καμμία σχέση με τη μεταβλητή Γ του κυρίου προγράμματος, η μία είναι τύπου Ακεραίου και η άλλη τύπου Πραγματικού. Αφού όλες οι μεταβλητές είναι

τοπικές, το ίδιο όνομα μεταβλητής μπορεί να εμφανίζεται σε διαφορετικά τμήματα προγράμματος χωρίς να αντιστοιχεί στην ίδια μεταβλητή.

Ό,τι ισχύει για τις μεταβλητές ισχύει και για τις σταθερές (συμβολικές).

Πολλές γλώσσες προγραμματισμού επιτρέπουν τη χρήση των μεταβλητών και των σταθερών, όχι μόνο στο τμήμα προγράμματος που δηλώνονται αλλά και σε άλλα ή ακόμη και σε όλα τα υπόλοιπα υποπρογράμματα. Αυτό που καθορίζει την περιοχή που ισχύουν οι μεταβλητές και οι σταθερές είναι η **εμβέλεια** των μεταβλητών της συγκεκριμένης γλώσσας προγραμματισμού.

Ορισμός: το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται εμβέλεια (scope) μεταβλητών.

Απεριόριστη εμβέλεια

Σύμφωνα με αυτή την αρχή όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν. Όλες οι μεταβλητές είναι **καθολικές**.

Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

Περιορισμένη εμβέλεια

Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος, να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι **τοπικές**, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν. Στη **ΓΛΩΣΣΑ** έχουμε περιορισμένη εμβέλεια.

Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων και η δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα μεταβλητής, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

Μερικώς περιορισμένη εμβέλεια

Σύμφωνα με αυτή την αρχή άλλες μεταβλητές είναι τοπικές και άλλες καθολικές. Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις που ορίζονται οι μεταβλητές ως τοπικές ή καθολικές.

Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά πλεονεκτήματα στον πεπειραμένο προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.

ΑΣΚΗΣΕΙΣ

ΣΥΝΑΡΤΗΣΕΙΣ

Παρατηρήσεις στον τρόπο κλήσης μιας συνάρτησης: Κατά την κλήση μιας συνάρτησης θα πρέπει οι πραγματικές παράμετροι να έχουν πριν πάρει κάποια τιμή (είτε με εντολή εκχώρησης είτε με την εντολή ΔΙΑΒΑΣΕ). Οι γενικές μορφές κλήσης μιας συνάρτησης είναι οι εξής:

1. Μεταβλητή ← όνομα_συνάρτησης(λίστα παραμέτρων).

Η μεταβλητή πρέπει να είναι του ίδιου τύπου με τον τύπο της συνάρτησης.

Παραδείγματα

α ← 16.8 β ← 19 ΜΟ ← Μέσος_Όρος(α, β)	ΜΟ ← Μέσος_Όρος(18, 15)	γ ← 12 δ ← 14 ΜΟ ← Μέσος_Όρος(γ, (γ+δ)/2)
---	-------------------------	---

2. Μεταβλητή ← Έκφραση. Η έκφραση περιλαμβάνει συνάρτηση.

Παραδείγματα

ΔΙΑΒΑΣΕ α β ← 9 γ ← 18 δ ← (α + MAX(α, β, γ)) / 2	ΔΙΑΒΑΣΕ α, β, γ max ← -MIN(-α, -β, -γ)
--	---

3. ΓΡΑΨΕ όνομα_συνάρτησης(λίστα παραμέτρων).

Παραδείγματα

ΔΙΑΒΑΣΕ α β ← 9 γ ← 18 ΓΡΑΨΕ MAX(α, β, γ)	ΔΙΑΒΑΣΕ α, β, γ ΓΡΑΨΕ -MIN(-α, -β, -γ)
--	---

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

1. Δίνεται το παρακάτω πρόγραμμα και οι συναρτήσεις που καλούνται. Τι θα εμφανίσει αν δοθεί σαν είσοδος η τιμή -2;

ΠΡΟΓΡΑΜΜΑ Παράδειγμα ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: α, γ ΑΡΧΗ ΔΙΑΒΑΣΕ α γ ← K(4, α) ΓΡΑΨΕ γ ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ	ΣΥΝΑΡΤΗΣΗ K(x, y): ΠΡΑΓΜΑΤΙΚΗ ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: α, x, y ΑΡΧΗ α ← x * y ΑΝ α > 0 ΤΟΤΕ K ← Δ(α) ΑΛΛΙΩΣ K ← Δ(-α) ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ	ΣΥΝΑΡΤΗΣΗ Δ(ρ): ΠΡΑΓΜΑΤΙΚΗ ΣΤΑΘΕΡΕΣ π = 3.14 ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: ρ ΑΡΧΗ Δ ← 2 * π * ρ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
---	--	--

ΛΥΣΗ

Οι εντολές με τη σειρά που εκτελούνται καθώς και σχολιασμός τους φαίνεται στον παρακάτω πίνακα:

ΔΙΑΒΑΣΕ α	Η μεταβλητή α παίρνει τιμή -2.
γ ← K(4, α)	Καλείται η συνάρτηση K. Οι τιμές 4 και -2 παίρνουν αντίστοιχα στις μεταβλητές x και y της συνάρτησης. Στη συνέχεια θα εκτελεστούν οι εντολές της συνάρτησης K.
α ← x * y	Στη μεταβλητή α της συνάρτησης (η οποία δεν έχει σχέση με τη μεταβλητή α του κυρίως προγράμματος) εκχωρείται η τιμή 4 * (-2) = -8.
ΑΝ α > 0 ΤΟΤΕ	Η συνθήκη είναι ψευδής, οπότε θα εκτελεστεί η εντολή μετά το ΑΛΛΙΩΣ .
K ← Δ(-α)	Καλείται η συνάρτηση Δ. Στη μεταβλητή ρ της συνάρτησης θα περαστεί η τιμή -α = 8. Στη συνέχεια θα εκτελεστούν οι εντολές της συνάρτησης Δ.
Δ ← 2 * π * ρ	Στο Δ καταχωρίζεται η τιμή 2 * 3.14 * 8 = 50.24
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ (της Δ)	Ολοκληρώνεται η εκτέλεση της συνάρτησης Δ και ο έλεγχος πηγαίνει στη συνάρτηση K που την κάλεσε. Με την εντολή K ← Δ(-α) το K θα πάρει τιμή 50.24.
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ (της K)	Ολοκληρώνεται η εκτέλεση της συνάρτησης K και ο έλεγχος πηγαίνει στο κυρίως πρόγραμμα που την κάλεσε. Με την εντολή γ ← K(4, α) το γ θα πάρει τιμή 50.24.
ΓΡΑΨΕ γ	εμφανίζεται η τιμή 50.24.

Ο πίνακας τιμών είναι ο παρακάτω: Τα γκρι πλαίσια δείχνουν τι εκτελείται σε κάθε εντολή:

Εντολή	Κύριο Πρόγραμμα		Συνάρτηση K				Συνάρτηση Δ	
	α	γ	α	κ	γ	Κ	ρ	Δ
ΔΙΑΒΑΣΕ α	-2							
$\gamma \leftarrow K(4, \alpha)$				4	-2			
$\alpha \leftarrow x * y$			-8					
ΑΝ $\alpha > 0$ ΤΟΤΕ								
$K \leftarrow \Delta(-\alpha)$							8	
$\Delta \leftarrow 2 * \pi * \rho$								50.24
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ (της Δ)						50.24		
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ (της K)		50.24						
ΓΡΑΨΕ γ	στην οθόνη: 50.24							

2. Να γραφεί πρόγραμμα που θα διαβάζει 5 πραγματικούς αριθμούς και στη συνέχεια με τη βοήθεια συναρτήσεων:

α) Θα εντοπίζει το μεγαλύτερο από τους τρεις πρώτους αριθμούς.

β) Θα εντοπίζει το μεγαλύτερο από τους τρεις τελευταίους αριθμούς.

γ) Θα εντοπίζει το μεγαλύτερο από τους τρεις μεσαίους αριθμούς.

δ) Θα εμφανίζει το μικρότερο από τους τρεις αριθμούς που εντόπισε στα προηγούμενα ερωτήματα.

ΛΥΣΗ

ΠΡΟΓΡΑΜΜΑ Παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: α1, α2, α3, α4, α5, Α, Β, Γ, Δ

ΑΡΧΗ

ΔΙΑΒΑΣΕ α1, α2, α3, α4, α5

$A \leftarrow \text{MAX}(\alpha_1, \alpha_2, \alpha_3)$

$B \leftarrow \text{MAX}(\alpha_3, \alpha_4, \alpha_5)$

$\Gamma \leftarrow \text{MAX}(\alpha_2, \alpha_3, \alpha_4)$

$\Delta \leftarrow \text{MIN}(A, B, \Gamma)$

ΓΡΑΨΕ Δ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ MAX(α, β, γ): ΠΡΑΓΜΑΤΙΚΗ

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: α, β, γ

ΑΡΧΗ

$\text{MAX} \leftarrow \alpha$

ΑΝ $\beta > \text{MAX}$ ΤΟΤΕ

$\text{MAX} \leftarrow \beta$

ΤΕΛΟΣ_ΑΝ

ΑΝ $\gamma > \text{MAX}$ ΤΟΤΕ

$\text{MAX} \leftarrow \gamma$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΣΥΝΑΡΤΗΣΗ MIN(α, β, γ): ΠΡΑΓΜΑΤΙΚΗ

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: α, β, γ

ΑΡΧΗ

$\text{MIN} \leftarrow \alpha$

ΑΝ $\beta < \text{MIN}$ ΤΟΤΕ

$\text{MIN} \leftarrow \beta$

ΤΕΛΟΣ_ΑΝ

ΑΝ $\gamma < \text{MIN}$ ΤΟΤΕ

$\text{MIN} \leftarrow \gamma$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

3. Να γραφεί συνάρτηση η οποία θα δέχεται δύο θετικούς ακέραιους αριθμούς και θα ελέγχει αν ο πρώτος είναι ακέραιο πολλαπλάσιο του δεύτερου. Στη συνέχεια, να γράψετε πρόγραμμα που αρχικά θα διαβάζει δύο ακέραιους αριθμούς και χρησιμοποιώντας τη συνάρτηση θα ελέγχει αν ο πρώτος αριθμός είναι ακέραιο πολλαπλάσιο του δεύτερου και θα εμφανίζει κατάλληλο μήνυμα. Αν αυτό δεν ισχύει, θα ελέγχει αν ο δεύτερος αριθμός είναι ακέραιο πολλαπλάσιο του πρώτου εμφανίζοντας κατάλληλο μήνυμα και αν δεν ισχύει τίποτα από τα παραπάνω θα εμφανίζει «Κανένας αριθμός δεν είναι ακέραιο πολλαπλάσιο του άλλου».

ΛΥΣΗ

Η ζητούμενη συνάρτηση θα είναι λογικού τύπου και είναι η παρακάτω:

ΣΥΝΑΡΤΗΣΗ ΑκΠολ(α,β):**ΛΟΓΙΚΗ**

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α,β

ΑΡΧΗ

ΑΝ α mod β=0 **ΤΟΤΕ**

ΑκΠολ←**ΑΛΗΘΗΣ**

ΑΛΛΙΩΣ

ΑκΠολ←**ΨΕΥΔΗΣ**

ΤΕΛΟΣ ΑΝ

ΤΕΛΟΣ ΣΥΝΑΡΤΗΣΗΣ

Το κυρίως πρόγραμμα θα είναι το εξής:

ΠΡΟΓΡΑΜΜΑ Παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α,β

ΛΟΓΙΚΕΣ: ΑΠ1, ΑΠ2

ΑΡΧΗ

ΔΙΑΒΑΣΕ α,β

ΑΠ1←ΑκΠολ(α,β)

ΑΠ2←ΑκΠολ(β,α)

ΑΝ ΑΠ1=**ΑΛΗΘΗΣ** **ΤΟΤΕ**

ΓΡΑΨΕ 'Ο πρώτος αριθμός είναι ακέραιο πολλαπλάσιο του δεύτερου'

ΑΛΛΙΩΣ ΑΝ ΑΠ2=**ΑΛΗΘΗΣ** **ΤΟΤΕ**

ΓΡΑΨΕ 'Ο δεύτερος αριθμός είναι ακέραιο πολλαπλάσιο του πρώτου'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Κανένας αριθμός δεν είναι ακέραιο πολλαπλάσιο του άλλου'

ΤΕΛΟΣ ΑΝ

ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

4. Έστω η συνάρτηση $MO(a,b,\gamma)$ η οποία επιστρέφει το μέσο όρο τριών πραγματικών αριθμών. Να χαρακτηρίσετε τις παρακάτω εντολές του κυρίως προγράμματος ως σωστές ή λάθος:

α) $\Sigma \leftarrow MO(a,a,b)$ **β)** $MO \leftarrow MO(3,a,b)$ **γ)** $a \leftarrow MO(a,a,a)$ **δ)** $\lambda \leftarrow MO(MO(a,0,0),b,b)$

ε) $\sigma \leftarrow MO(a,b)$

5. Έστω το παρακάτω πρόγραμμα και συνάρτηση: Τι θα εμφανίσει αν δοθούν σαν είσοδο οι τιμές 4 και 1;

ΠΡΟΓΡΑΜΜΑ Υπολογισμός

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α,γ

ΑΡΧΗ

ΔΙΑΒΑΣΕ α,γ

$\gamma \leftarrow \text{Συναρτ}(a, 2*\gamma) + \text{Συναρτ}(\gamma, 2)$

ΓΡΑΨΕ γ

ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Συναρτ(κ,λ):**ΑΚΕΡΑΙΑ**

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: κ,λ,μ

ΑΡΧΗ

ΑΝ κ>λ **ΤΟΤΕ**

$\mu \leftarrow \kappa * \lambda$

ΑΛΛΙΩΣ

$\mu \leftarrow \kappa + \lambda - 2$

ΤΕΛΟΣ ΑΝ

Συναρτ← $\mu * 2$

ΤΕΛΟΣ ΣΥΝΑΡΤΗΣΗΣ

6. Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
ΠΡΟΓΡΑΜΜΑ Άσκηση  
ΜΕΤΑΒΛΗΤΕΣ  
ΑΚΕΡΑΙΕΣ: α, β  
ΑΡΧΗ  
  α ← 5  
  β ← 1  
  α ← Συναρ(α, β)  
  α ← Συναρ(β, α)  
  β ← Συναρ(β, α)  
  ΓΡΑΨΕ α, β  
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
ΣΥΝΑΡΤΗΣΗ Συναρ(κ, π): ΑΚΕΡΑΙΑ  
ΜΕΤΑΒΛΗΤΕΣ  
ΑΚΕΡΑΙΕΣ: κ, π  
ΑΡΧΗ  
  ΟΣΟ κ > π ΕΠΑΝΑΛΑΒΕ  
    κ ← κ - 3  
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
  Συναρ ← κ  
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

7. Έστω το παρακάτω κύριο πρόγραμμα και οι συναρτήσεις που καλούνται. Τι θα εμφανίσει αν σαν είσοδος δοθεί η τιμή 4.5;

```
ΠΡΟΓΡΑΜΜΑ Υπολογισμός  
ΜΕΤΑΒΛΗΤΕΣ  
ΠΡΑΓΜΑΤΙΚΕΣ: α, γ  
ΑΡΧΗ  
  ΔΙΑΒΑΣΕ α  
  γ ← Συναρ1(α)  
  γ ← Συναρ1(γ + α)  
  ΓΡΑΨΕ γ  
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
ΣΥΝΑΡΤΗΣΗ Συναρ1(ρ): ΠΡΑΓΜΑΤΙΚΗ  
ΜΕΤΑΒΛΗΤΕΣ  
ΠΡΑΓΜΑΤΙΚΕΣ: ρ  
ΑΚΕΡΑΙΕΣ: β  
ΑΡΧΗ  
  β ← A_M(ρ)  
  ΑΝ ρ = β ΤΟΤΕ  
    Συναρ1 ← ρ + β  
  ΑΛΛΙΩΣ  
    Συναρ1 ← Συναρ2(ρ, ρ)  
  ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

```
ΣΥΝΑΡΤΗΣΗ Συναρ2(x, y): ΠΡΑΓΜΑΤΙΚΗ  
ΜΕΤΑΒΛΗΤΕΣ  
ΠΡΑΓΜΑΤΙΚΕΣ: x, y  
ΑΡΧΗ  
  ΑΝ x = y ΤΟΤΕ  
    Συναρ2 ← x + y  
  ΑΛΛΙΩΣ  
    Συναρ2 ← 2 * x ^ 2 + y  
  ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

8. Δίνεται το παρακάτω πρόγραμμα:

```
ΠΡΟΓΡΑΜΜΑ Εξίσωση  
ΜΕΤΑΒΛΗΤΕΣ  
ΑΚΕΡΑΙΕΣ: είδος  
ΠΡΑΓΜΑΤΙΚΕΣ: α, β  
ΑΡΧΗ  
  ΔΙΑΒΑΣΕ α, β  
  είδος ← Πρωτοβάθμια(α, β)  
  ΑΝ είδος = 1 ΤΟΤΕ  
    ΓΡΑΨΕ 'Αδύνατη'  
  ΑΛΛΙΩΣ_ΑΝ είδος = 2 ΤΟΤΕ  
    ΓΡΑΨΕ 'Αόριστη'  
  ΑΛΛΙΩΣ  
    ΓΡΑΨΕ -β/α  
  ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Το πρόγραμμα αυτό διαβάζει τους συντελεστές α και β της πρωτοβάθμιας εξίσωσης $ax + b = 0$ και με τη βοήθεια της συνάρτησης Πρωτοβάθμια, υπολογίζει τη λύση της. Να υλοποιήσετε τη συνάρτηση Πρωτοβάθμια η οποία δέχεται τους συντελεστές της εξίσωσης και επιστρέφει 1 αν αυτή είναι αδύνατη, 2 αν είναι αόριστη και 3 αν έχει λύση.

9. Στους ημιτελικούς ενός διαγωνισμού που τραγουδιού που πραγματοποιείται σε 20 δήμους της χώρας έχουν προκριθεί 3 υποψήφιοι από κάθε δήμο. Στον τελικό θα περάσει από κάθε δήμο ο υποψήφιος με το μεγαλύτερο βαθμό. Να γραφεί πρόγραμμα που θα διαβάσει τα ονόματα και τους βαθμούς των υποψηφίων από κάθε δήμο και θα εμφανίζει τα ονόματα αυτών που πέρασαν στον τελικό. Να χρησιμοποιήσετε συνάρτηση την οποία θα φτιάξετε που θα δέχεται τους βαθμούς και τα ονόματα 3 υποψηφίων και θα επιστρέφει το όνομα αυτού με τον μεγαλύτερο βαθμό.

10. Για την εισαγωγή σε ένα μεταπτυχιακό τμήμα μιας πανεπιστημιακής σχολής, οι φοιτητές διαγωνίζονται σε 3 μαθήματα στα οποία αθροιστικά θα πρέπει να συγκεντρώσουν τουλάχιστον 17 βαθμούς. Το άριστα σε κάθε μάθημα είναι 10. Πριν τον υπολογισμό της συνολικής βαθμολογίας οι βαθμοί στρογγυλοποιούνται προς τα πάνω στη περίπτωση που το δεκαδικό τους μέρος είναι μεγαλύτερο του 0.5. Π.χ αν κάποιος φοιτητής

έχει σε ένα μάθημα βαθμό 7.6 τότε ο βαθμός θα γίνει 8. Αν ο βαθμός είναι π.χ. 7.4 τότε παραμένει αμετάβλητος.

α) Να υλοποιηθεί συνάρτηση η οποία θα κάνει τη στρογγυλοποίηση του βαθμού που περιγράφηκε παραπάνω.

β) Να γραφεί πρόγραμμα που θα δέχεται τους 3 βαθμούς ενός φοιτητή και με τη χρήση της προηγούμενης συνάρτησης θα εμφανίζει κατάλληλο μήνυμα για το αν ο φοιτητής περνά στο μεταπτυχιακό τμήμα ή όχι.

ΔΙΑΔΙΚΑΣΙΕΣ

Παρατηρήσεις στον τρόπο κλήσης μιας διαδικασίας: Κατά την κλήση μιας διαδικασίας δεν επιτρέπεται στη λίστα των πραγματικών παραμέτρων της (σε αντίθεση με τις συναρτήσεις) να υπάρχει κάποια σταθερά ή κάποια έκφραση. Επιτρέπεται μόνο η χρήση μεταβλητών ακόμα κι'αν αυτές δεν έχουν πάρει τιμή (σε αντίθεση με τις συναρτήσεις που θα πρέπει στη λίστα των πραγματικών παραμέτρων τους να υπάρχουν σταθερές ή μεταβλητές που έχουν πάρει τιμή ή ακόμα και εκφράσεις). Αυτό συμβαίνει γιατί κατά την επιστροφή από τη διαδικασία στο πρόγραμμα που την κάλεσε γίνεται και μεταβίβαση των τιμών των τυπικών παραμέτρων στις αντίστοιχες πραγματικές της διαδικασίας (δες παρ. 10.5.3 σελ.6). Κάτι τέτοιο δεν συμβαίνει στις συναρτήσεις.

Παραδείγματα σωστής κλήσης διαδικασίας: ΚΑΛΕΣΕ ΜΟ(α,β,γ)

ΚΑΛΕΣΕ ΕΚΤΥΠΩΣΗ()

Παραδείγματα λανθασμένης κλήσης διαδικασίας: ΚΑΛΕΣΕ ΜΟ(17,β,γ)

ΚΑΛΕΣΕ ΜΟ(α, β+γ,δ)

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

11. Δίνεται το παρακάτω πρόγραμμα και η διαδικασία που καλείται από αυτό.

ΠΡΟΓΡΑΜΜΑ Παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α,β

ΠΡΑΓΜΑΤΙΚΕΣ: γ,δ

ΑΡΧΗ

ΔΙΑΒΑΣΕ α,β

ΚΑΛΕΣΕ Διαδ(α,β,γ,δ)

ΓΡΑΨΕ α,γ,δ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Διαδ(κ,α,μ,ν)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: κ,α

ΠΡΑΓΜΑΤΙΚΕΣ: μ,ν,λ

ΑΡΧΗ

λ←κ-α

ΑΝ λ>0 **ΤΟΤΕ**

μ←α-κ

ν←α*κ

ΑΛΛΙΩΣ

μ←α+κ

ν←α*κ

κ←λ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

α) ποιες είναι οι πραγματικές και οι τυπικές παράμετροι;

β) τι θα εμφανίσει αν δοθούν σαν είσοδος οι τιμές 2 και 8;

ΛΥΣΗ

α) Οι πραγματικές παράμετροι είναι οι α,β,γ,δ και οι τυπικές παράμετροι είναι οι κ,α,μ,ν. Η μεταβλητή α που βρίσκεται στη λίστα τυπικών παραμέτρων δεν έχει σχέση με τη μεταβλητή α του κυρίως προγράμματος γιατί πρόκειται για διαφορετική μεταβλητή (στην ουσία κατέχει διαφορετική θέση στη μνήμη του υπολογιστή).

β) Οι εντολές με τη σειρά που εκτελούνται καθώς και σχολιασμός τους φαίνεται στον παρακάτω πίνακα:

ΔΙΑΒΑΣΕ α,β	Οι τιμές 2 και 8 περνούν στις μεταβλητές του κυρίου προγράμματος α και β αντίστοιχα.
ΚΑΛΕΣΕ Διαδ(α,β,γ,δ)	Γίνεται κλήση της διαδικασίας και οι τιμές των α και β του κυρίως προγράμματος μεταβιβάζονται στις κ και α αντίστοιχα της διαδικασίας δηλ. κ=2 και α=8. (στις μ και ν δεν μεταβιβάζεται κάποια τιμή γιατί οι γ και δ δεν έχουν πάρει κάποια τιμή στο κυρίως πρόγραμμα)
λ←κ-α	λ=2-8=-6
ΑΝ λ>0 ΤΟΤΕ	δεν ισχύει άρα πάμε στις εντολές μετά το ΑΛΛΙΩΣ
μ←α+κ	μ=8+2=10
ν←α*κ	ν=8*2=16
κ←λ	κ=-6

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	τελειώνει η διαδικασία και μεταβιβάζονται οι τιμές των τυπικών παραμέτρων στις αντίστοιχες πραγματικές δηλαδή: $\kappa \rightarrow \alpha = -6$ $\alpha \rightarrow \beta = 8$ $\mu \rightarrow \gamma = 10$ $\nu \rightarrow \delta = 16$
ΓΡΑΨΕ α,γ,δ	εμφανίζονται οι τιμές -6,10,16.

Παρακάτω φαίνεται και ο πίνακας τιμών κατά την εκτέλεση:

Εντολές	Κύριο πρόγραμμα				Διαδικασία				
	α	β	γ	δ	κ	α	μ	ν	λ
ΔΙΑΒΑΣΕ α,β	2	8							
ΚΑΛΕΣΕ Διαδ(α,β,γ,δ)					2	8			
$\lambda \leftarrow \kappa - \alpha$									-6
ΑΝ $\lambda > 0$ ΤΟΤΕ									
$\mu \leftarrow \alpha + \kappa$							10		
$\nu \leftarrow \alpha * \kappa$								16	
$\kappa \leftarrow \lambda$					-6				
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	-6	8	10	16					
ΓΡΑΨΕ α,γ,δ	εμφανίζεται -6,10,16								

12. Να γίνουν οι εξής διαδικασίες:

α) Διαδικασία που θα δέχεται την ακτίνα ενός κύκλου και θα επιστρέφει το εμβαδό και την περίμετρό του.

β) Διαδικασία που θα δέχεται την ακτίνα ενός κύκλου και θα εμφανίζει το εμβαδό και την περίμετρό του.

γ) Διαδικασία που θα διαβάζει την ακτίνα ενός κύκλου και θα επιστρέφει το εμβαδό και την περίμετρό του.

δ) Διαδικασία που θα διαβάζει την ακτίνα ενός κύκλου και θα εμφανίζει το εμβαδό και την περίμετρό του.

Δίνεται ότι $E = \pi r^2$ και $\Pi = 2\pi r$, όπου E =το εμβαδό, r =η ακτίνα και Π =η περίμετρος του κύκλου.

ΛΥΣΗ

<u>α.</u> ΔΙΑΔΙΚΑΣΙΑ ΚΥΚΛΟΣ1(ρ,Ε,ΠΕΡ) ΣΤΑΘΕΡΕΣ $\pi = 3.14$ ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: ρ,Ε,ΠΕΡ ΑΡΧΗ $E \leftarrow \pi * \rho^2$ $\Pi \leftarrow 2 * \pi * \rho$ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	<u>β.</u> ΔΙΑΔΙΚΑΣΙΑ ΚΥΚΛΟΣ2(ρ) ΣΤΑΘΕΡΕΣ $\pi = 3.14$ ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: ρ,Ε,ΠΕΡ ΑΡΧΗ $E \leftarrow \pi * \rho^2$ $\Pi \leftarrow 2 * \pi * \rho$ ΓΡΑΨΕ Ε,ΠΕΡ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
<u>γ.</u> ΔΙΑΔΙΚΑΣΙΑ ΚΥΚΛΟΣ3(Ε,ΠΕΡ) ΣΤΑΘΕΡΕΣ $\pi = 3.14$ ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: ρ,Ε,ΠΕΡ ΑΡΧΗ ΔΙΑΒΕΣΕ ρ $E \leftarrow \pi * \rho^2$ $\Pi \leftarrow 2 * \pi * \rho$ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ	<u>δ.</u> ΔΙΑΔΙΚΑΣΙΑ ΚΥΚΛΟΣ4() ΣΤΑΘΕΡΕΣ $\pi = 3.14$ ΜΕΤΑΒΛΗΤΕΣ ΠΡΑΓΜΑΤΙΚΕΣ: ρ,Ε,ΠΕΡ ΑΡΧΗ ΔΙΑΒΑΣΕ ρ $E \leftarrow \pi * \rho^2$ $\Pi \leftarrow 2 * \pi * \rho$ ΓΡΑΨΕ Ε,ΠΕΡ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

- α.** Η διαδικασία ΚΥΚΛΟΣ1 έχει 3 παραμέτρους, μία για να δέχεται την ακτίνα και δύο για να επιστρέφει το εμβαδό και την περίμετρο.
- β.** Η διαδικασία ΚΥΚΛΟΣ2 έχει μία παράμετρο για να δέχεται την ακτίνα, αφού το εμβαδό και την περίμετρο τα εμφανίζει μέσα στη διαδικασία.
- γ.** Η διαδικασία ΚΥΚΛΟΣ3 έχει δύο παραμέτρους για να επιστρέφει το εμβαδό και την περίμετρο αφού την ακτίνα τη διαβάσει μέσα της.
- δ.** Η διαδικασία ΚΥΚΛΟΣ4 δεν έχει καμία παράμετρο αφού σύμφωνα με την εκφώνηση δεν επιστρέφει κάτι ούτε δέχεται κάτι από το κυρίως πρόγραμμα.
- Σημείωση:** Πρέπει να είμαστε προσεκτικοί στην εκφώνηση: όταν μας λέει ότι η διαδικασία θα δέχεται π.χ. μία τιμή εννοείται ότι την τιμή θα τη δέχεται όχι από το χρήστη αλλά από το κυρίως πρόγραμμα ή υποπρόγραμμα που την καλεί, μέσω παραμέτρου. Επίσης όταν μας λέει ότι θα επιστρέφει μία τιμή, την τιμή θα την επιστρέφει στο πρόγραμμα που την κάλεσε μέσω παραμέτρου.

13. Ο τελικός βαθμός των μαθητών ενός σχολείου υπολογίζεται με βάση δύο γραπτούς και δύο προφορικούς βαθμούς. Οι δύο γραπτοί βαθμοί δεν πρέπει να διαφέρουν μεταξύ τους πάνω από δύο μονάδες. Αν συμβαίνει αυτό, τότε ο μεγαλύτερος από τους δύο βαθμούς μειώνεται προς τα κάτω ώστε να διαφέρει ακριβώς δύο μονάδες από τον μικρότερο. Το ίδιο ισχύει και για τους προφορικούς βαθμούς. Μετά την αναπροσαρμογή των βαθμών ο τελικός βαθμός είναι ο μέσος όρος των τεσσάρων βαθμών. Να γραφεί πρόγραμμα που θα διαβάσει τους τέσσερις βαθμούς ενός μαθητή και αφού κάνει αναπροσαρμογή των βαθμών με τη βοήθεια διαδικασίας, θα εμφανίζει το τελικό βαθμό του μαθητή.

ΛΥΣΗ

ΠΡΟΓΡΑΜΜΑ Βαθμοί
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: γρ1,γρ2,προφ1,προφ2,τελ_βαθ
ΑΡΧΗ
ΔΙΑΒΑΣΕ γρ1,γρ2,προφ1,προφ2
ΚΑΛΕΣΕ ΑΝΑΠΡΟΣ(γρ1,γρ2)
ΚΑΛΕΣΕ ΑΝΑΠΡΟΣ(προφ1,προφ2)
 τελ_βαθ ← (γρ1+γρ2+προφ1+προφ2)/4
ΓΡΑΨΕ τελ_βαθ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΠΡΟΣ(ΒΑΘ1,ΒΑΘ2)
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: ΒΑΘ1,ΒΑΘ2
ΑΡΧΗ
ΑΝ ΒΑΘ1-ΒΑΘ2>2 **ΤΟΤΕ**
 ΒΑΘ1 ← ΒΑΘ2+2
ΑΛΛΙΩΣ_ΑΝ ΒΑΘ2-ΒΑΘ1>2 **ΤΟΤΕ**
 ΒΑΘ2 ← ΒΑΘ1+2
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

14. Δίνεται το παρακάτω πρόγραμμα:

ΠΡΟΓΡΑΜΜΑ Κύριο
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: Α,Β,Γ
ΑΡΧΗ
ΔΙΑΒΑΣΕ Α,Β,Γ
ΚΑΛΕΣΕ Διαδ1(Α,Β,Γ)
ΓΡΑΨΕ Α,Β,Γ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Διαδ1(Β,Α,Γ)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: Α,Β,Γ
ΑΡΧΗ
 Α ← Α+2
 Β ← Β-3
 Γ ← Α+Β
ΓΡΑΨΕ Α,Β,Γ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Τι θα εμφανιστεί κατά την εκτέλεση του προγράμματος αν ως τιμές εισόδου δοθούν οι αριθμοί 5,7,10;

15. Δίνεται το παρακάτω πρόγραμμα:

ΠΡΟΓΡΑΜΜΑ Κλήση_Υποπρογ
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: α,β,χ
ΑΡΧΗ
 α ← 1
 β ← 2
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ α<=4 **ΤΟΤΕ**
ΚΑΛΕΣΕ Διαδ1(α,β,χ)
ΑΛΛΙΩΣ
 χ ← Συν1(α,β)
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ α,β,χ
ΜΕΧΡΙΣ_ΟΤΟΥ χ>11
ΓΡΑΨΕ χ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Διαδ1(λ,κ,μ)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: κ,λ,μ
ΑΡΧΗ
 κ ← κ+1
 λ ← λ+3
 μ ← κ+λ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΣΥΝΑΡΤΗΣΗ Συν1(ε,ζ):**ΑΚΕΡΑΙΑ**
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: ε,ζ
ΑΡΧΗ
 ζ ← ζ+2
 ε ← ε*2
 Συν1 ← ε+ζ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Να γράψετε στο τετράδιό σας τις τιμές που θα εμφανιστούν κατά την εκτέλεση του προγράμματος.

16. Τι θα εμφανίσει το παρακάτω πρόγραμμα αν εκτελεστεί με είσοδο τις τιμές -2 και 4;

ΠΡΟΓΡΑΜΜΑ Υπολογισμός
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: α,β,γ
ΑΡΧΗ
 ΔΙΑΒΑΣΕ α,β
 $\gamma \leftarrow \Sigma(\alpha, \beta)$
 ΚΑΛΕΣΕ Δ(α,β,γ)
 ΓΡΑΨΕ α,β,γ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Δ(κ,λ,μ)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: κ,λ,μ,α
ΑΡΧΗ
 $\alpha \leftarrow \mu$
 $\mu \leftarrow \kappa$
 $\kappa \leftarrow \alpha$
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΣΥΝΑΡΤΗΣΗ Σ(α,β):**ΑΚΕΡΑΙΑ**
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: α,β,γ
ΑΡΧΗ
 ΑΝ $\beta * \alpha \leq 0$ **ΤΟΤΕ**
 $\gamma \leftarrow \beta$
 ΚΑΛΕΣΕ Δ(α,β,γ)
 ΑΛΛΙΩΣ
 $\gamma \leftarrow \alpha$
 ΚΑΛΕΣΕ Δ(α,β,γ)
 ΤΕΛΟΣ_ΑΝ
 $\Sigma \leftarrow \alpha$
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

17. Δίνεται το παρακάτω πρόγραμμα:

ΠΡΟΓΡΑΜΜΑ Υπολογισμοί
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: α,β,γ
ΑΡΧΗ
 ΔΙΑΒΑΣΕ α,β
 $\gamma \leftarrow \alpha + \text{Πράξη}(\alpha, \beta)$
 ΓΡΑΨΕ γ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Πράξη(κ,γ):**ΠΡΑΓΜΑΤΙΚΗ**
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: κ, γ
ΑΡΧΗ
 ΑΝ $\kappa \geq \gamma$ **ΤΟΤΕ**
 $\text{Πράξη} \leftarrow \kappa - \gamma$
 ΑΛΛΙΩΣ
 $\text{Πράξη} \leftarrow \kappa + \gamma$
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

α. Να ξαναγράψετε το πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία χρησιμοποιώντας διαδικασία αντί συνάρτησης, την οποία διαδικασία και να κατασκευάσετε.

β. Να ξαναγράψετε το πρόγραμμα που δόθηκε αρχικά, ώστε να επιτελεί την ίδια λειτουργία χωρίς τη χρήση υποπρογράμματος.

γ. Να γράψετε στο τετράδιό σας τις τιμές που θα εμφανιστούν κατά την εκτέλεση του αρχικού προγράμματος που δόθηκε, αν ως τιμές εισόδου δοθούν οι αριθμοί:

i) α=10, β=5

ii) α=5, β=5

iii) α=3, β=5

18. Να γίνει πρόγραμμα που θα διαβάζει τον γραπτό και προφορικό βαθμό ενός μαθητή. Ο προφορικός βαθμός πρέπει να είναι μέχρι 3 μονάδες μεγαλύτερος από το γραπτό. Αν αυτό δεν ισχύει, τότε ο προφορικός βαθμός αναπροσαρμόζεται ώστε να διαφέρει ακριβώς 3 μονάδες από το γραπτό. Να γραφεί υποπρόγραμμα με όνομα Μέσος, το οποίο θα δέχεται τον γραπτό και προφορικό βαθμό, θα κάνει την αναπροσαρμογή του προφορικού αν χρειάζεται και θα επιστρέφει το μέσο όρο των δύο βαθμών. Στο κυρίως πρόγραμμα να εμφανίζεται ο μέσος όρος πριν και μετά την αναπροσαρμογή.

Δύο μαθητές έδωσαν τις παρακάτω λύσεις:

A.

ΠΡΟΓΡΑΜΜΑ Βαθμός
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: γρ,πρ,ΜΟ,ΜΟ_α
ΑΡΧΗ
 ΔΙΑΒΑΣΕ γρ,πρ
 $\text{ΜΟ_}\alpha \leftarrow \text{Μέσος}(\gamma\rho, \pi\rho)$
 $\text{ΜΟ} \leftarrow (\gamma\rho + \pi\rho) / 2$
 ΓΡΑΨΕ ΜΟ,ΜΟ_α
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Μέσος(γ,π): **ΠΡΑΓΜΑΤΙΚΗ**
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: γ,π
ΑΡΧΗ
 ΑΝ $(\pi - \gamma) > 3$ **ΤΟΤΕ**
 $\pi \leftarrow \gamma + 3$
 ΤΕΛΟΣ_ΑΝ
 $\text{Μέσος} \leftarrow (\gamma + \pi) / 2$
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

B.

ΠΡΟΓΡΑΜΜΑ Βαθμός
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: γρ,πρ,ΜΟ,ΜΟ_α
ΑΡΧΗ
 ΔΙΑΒΑΣΕ γρ,πρ
 ΚΑΛΕΣΕ Μέσος(γρ,πρ,ΜΟ_α)
 $\text{ΜΟ} \leftarrow (\gamma\rho + \pi\rho) / 2$
 ΓΡΑΨΕ ΜΟ,ΜΟ_α
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Μέσος(γ,π,ΜΟ_α)
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ: γ,π,ΜΟ_α
ΑΡΧΗ
 ΑΝ $(\pi - \gamma) > 3$ **ΤΟΤΕ**
 $\pi \leftarrow \gamma + 3$
 ΤΕΛΟΣ_ΑΝ
 $\text{ΜΟ_}\alpha \leftarrow (\gamma + \pi) / 2$
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Είναι σωστές οι δύο λύσεις; Δικαιολογήστε την απάντησή σας.

19. Σε ένα εκπαιδευτικό σύστημα κάθε μαθητής εξετάζεται σε δύο γραπτές και μια προφορική εξέταση. Στην περίπτωση που ο προφορικός βαθμός διαφέρει λιγότερο από 3 μονάδες από το μέσο όρο των γραπτών βαθμών, τότε ο τελικός βαθμός των μαθητών υπολογίζεται ως ο μέσος όρος και των τριών βαθμολογιών. Σε διαφορετική περίπτωση ο προφορικός βαθμός αγνοείται και ο τελικός βαθμός υπολογίζεται ως ο μέσος όρος μόνο των γραπτών βαθμών.

i. Να γράψετε διαδικασία η οποία θα δέχεται τους τρεις βαθμούς ενός μαθητή και θα υπολογίζει και θα επιστρέφει το τελικό του βαθμό. Επιπλέον, θα εμφανίζει κατάλληλο μήνυμα σε περίπτωση που αγνοήθηκε ο προφορικός βαθμός.

ii. Να γράψετε κύριο πρόγραμμα που θα διαβάζει τους γραπτούς και τον προφορικό βαθμό 30 μαθητών μιας τάξης και καλώντας το υποπρόγραμμα του ερωτήματος (i) θα εμφανίζει τον τελικό τους βαθμό.

20. Να αναπτύξετε πρόγραμμα που με τη βοήθεια διαδικασιών που θα φτάξετε θα κάνει τα παρακάτω:

α. Θα διαβάζει μια θερμοκρασία εκφρασμένη σε βαθμούς Κελσίου. Η θερμοκρασία που θα διαβαστεί θα πρέπει να είναι μεγαλύτερη του απόλυτου μηδενός (-273°C) για να είναι έγκυρη. Σε περίπτωση λανθασμένης εισόδου θα εμφανίζεται κατάλληλο μήνυμα και θα ζητείται ξανά η τιμή.

β. Θα μετατρέπει τη θερμοκρασία αυτή σε βαθμούς Φαρενάιτ σύμφωνα με τον τύπο:

$$\frac{F-32}{9} = \frac{C}{5}$$

γ. Θα τυπώνει το αποτέλεσμα με τη μορφή: « Οι ____ βαθμοί Κελσίου αντιστοιχούν σε ____ βαθμούς Φαρενάιτ ».

21. Ποιο από τα παρακάτω υποπρογράμματα μπορεί να υλοποιηθεί με συνάρτηση;

α. Θα δέχεται τις δύο βάσεις και το ύψος ενός τραπεζίου και θα επιστρέφει το εμβαδόν του.

β. Θα διαβάζει το προφορικό και γραπτό βαθμό ενός μαθητή και θα επιστρέφει το μέσο όρο του.

γ. Θα δέχεται 3 αριθμούς και θα εμφανίζει το μεγαλύτερο.

δ. Θα δέχεται 3 αριθμούς και θα επιστρέφει το άθροισμά τους.

ε. Θα δέχεται δύο ακέραιες τιμές και θα ελέγχει αν και οι δύο είναι περιττές.

στ. Θα υπολογίζει τους δύο μικρότερους από πέντε ακέραιους.

ζ. Ελέγχει αν ένας χαρακτήρας είναι φωνήεν ή σύμφωνο.

22.i) Να γράψετε συνάρτηση η οποία θα δέχεται έναν ακέραιο αριθμό n και θα υπολογίζει την παράσταση: $1+2+3+\dots+n$.

ii) Να γράψετε διαδικασία που θα κάνει την ίδια λειτουργία.

iii) Να γράψετε δύο προγράμματα που θα διαβάζουν μία ακέραια θετική τιμή n και το πρώτο με χρήση της συνάρτησης και το δεύτερο με χρήση της διαδικασίας θα εμφανίζουν την τιμή της παράστασης:

$$Y = \frac{(1+2+3+\dots+50)^2 + (1+2+3+\dots+n)^3}{(1+2+3+\dots+2n)^2}$$

ΧΡΗΣΗ ΠΙΝΑΚΩΝ ΣΤΑ ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

- Στη λίστα παραμέτρων μιας διαδικασίας ή συνάρτησης μπορούν να υπάρχουν και πίνακες μονοδιάστατοι ή διοδιάστατοι.
- Το μέγεθος του πίνακα ορίζεται στο τμήμα δηλώσεων του υποπρογράμματος και όχι στη λίστα παραμέτρων του. π.χ. αν μια διαδικασία με όνομα Διαδ χρησιμοποιεί έναν πίνακα $\Pi[10,20]$, τότε στο κυρίως πρόγραμμα θα την καλέσουμε ως εξής:

ΚΑΛΕΣΕ Διαδ(Π) και όχι **ΚΑΛΕΣΕ** Διαδ($\Pi[10,20]$).

- Οι πίνακες που υπάρχουν στη λίστα των πραγματικών παραμέτρων μιας διαδικασίας ή συνάρτησης πρέπει να είναι του ίδιου τύπου και του ίδιου μεγέθους με τους αντίστοιχους πίνακες στη λίστα των τυπικών παραμέτρων.
- Οι συναρτήσεις μπορούν να δέχονται αλλά όχι να επιστρέφουν πίνακα.

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

23. Σε έναν αγώνα σκοποβολής, κάθε αθλητής ρίχνει 10 βολές και η τελική του βαθμολογία υπολογίζεται ως ο μέσος όρος των βολών του εκτός της μεγαλύτερης και μικρότερης. Να γραφεί πρόγραμμα που θα δέχεται τους βαθμούς των 10 βολών για 30 αθλητές και θα υπολογίζει και εμφανίζει την τελική βαθμολογία του κάθε αθλητή. Η υλοποίηση του προγράμματος να γίνει με χρήση μίας διαδικασίας που θα διαβάζει τους βαθμούς των 10 βολών σε έναν πίνακα 10 θέσεων και μίας δεύτερης διαδικασίας που θα βρίσκει το μέγιστο και ελάχιστο στοιχείο ενός πίνακα 10 θέσεων.

ΛΥΣΗ

Η πρώτη διαδικασία θα δέχεται έναν πίνακα 10 θέσεων (χωρίς τιμές) και θα τον επιστρέφει διαβασμένο:

ΔΙΑΔΙΚΑΣΙΑ Εισαγωγή(ΒΟΛΕΣ)

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ΒΟΛΕΣ[10]

ΑΚΕΡΑΙΕΣ: i

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

ΔΙΑΒΑΣΕ ΒΟΛΕΣ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Η δεύτερη διαδικασία θα δέχεται έναν πίνακα 10 θέσεων (10 βολές ενός αθλητή) και θα επιστρέφει το ελάχιστο και μέγιστο στοιχείο του πίνακα:

ΔΙΑΔΙΚΑΣΙΑ MAX_MIN(ΠΙΝ,ΜΕΓ,ΕΛΑΧ)

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ΠΙΝ[10],ΜΕΓ,ΕΛΑΧ

ΑΚΕΡΑΙΕΣ: i

ΑΡΧΗ

ΜΕΓ←ΠΙΝ[1]

ΕΛΑΧ←ΠΙΝ[1]

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10

ΑΝ ΠΙΝ[i]>ΜΕΓ ΤΟΤΕ

ΜΕΓ←ΠΙΝ[i]

ΤΕΛΟΣ_ΑΝ

ΑΝ ΠΙΝ[i]<ΕΛΑΧ ΤΟΤΕ

ΕΛΑΧ←ΠΙΝ[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Το κυρίως πρόγραμμα που θα καλεί τις δύο διαδικασίες θα είναι:

ΠΡΟΓΡΑΜΜΑ Σκοποβολή

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, κ

ΠΡΑΓΜΑΤΙΚΕΣ: ΒΟΛΕΣ[10],Μ,Ε,ΑΘΡ,ΤΕΛ_ΒΑΘ

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30

ΚΑΛΕΣΕ Εισαγωγή(ΒΟΛΕΣ)

ΚΑΛΕΣΕ MAX_MIN(ΒΟΛΕΣ,Μ,Ε) !Κατά την επιστροφή από τη διαδικασία το Μ θα περιέχει το

ΑΘΡ←0

ΓΙΑ κ ΑΠΟ 1 ΜΕΧΡΙ 10
ΑΘΡ←ΑΘΡ+ΒΟΛΕΣ[κ]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΘΡ←ΑΘΡ-Μ-Ε

ΤΕΛ_ΒΑΘ←ΑΘΡ/8

ΓΡΑΨΕ ΤΕΛ_ΒΑΘ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

!για κάθε αθλητή από τον 1^ο ως τον 30^ο.

!Κατά την επιστροφή από τη διαδικασία ο πίνακας θα περιέχει τιμές μέγιστο και το Ε το ελάχιστο στοιχείο του πίνακα με τις βολές.

} υπολογισμός του αθροίσματος των βολών για κάθε αθλητή

!υπολογισμός της τελικής βαθμολογίας κάθε αθλητή.

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

24. Δένεται η επόμενη διαδικασία:

ΔΙΑΔΙΚΑΣΙΑ Άσκηση(Α,κ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i,j,κ,μ,temp,A[10]

ΑΡΧΗ

μ←0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ κ

μ←μ+1

A[μ]←i

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ μ ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ -1

ΑΝ A[j-1]<A[j] ΤΟΤΕ

temp←A[j]

A[j]←A[j-1]

A[j-1]←temp

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ $\kappa=2$ ΤΟΤΕ

$\kappa \leftarrow 4$

$A[2] \leftarrow A[\kappa]$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Τι τιμές θα έχει η μεταβλητή κ και ο πίνακας B μετά την εκτέλεση των παρακάτω εντολών;

1) $\kappa \leftarrow 3$

ΚΑΛΕΣΕ Άσκηση(B, κ)

2) $\kappa \leftarrow 2$

ΚΑΛΕΣΕ Άσκηση(B, κ)

25. Δίνεται η παρακάτω διαδικασία:

ΔΙΑΔΙΚΑΣΙΑ Πίνακας(A, κ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: $i, \kappa, temp, A[6]$

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 3

ΑΝ $A[i] < \kappa$ ΤΟΤΕ

$temp \leftarrow A[i]$

$A[i] \leftarrow A[6-i+1]$

$A[6-i+1] \leftarrow temp$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6

ΓΡΑΨΕ $A[i]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Αν ο πίνακας A έχει τα στοιχεία 15, 1, 2, 9, 22, 16 τι θα εμφανιστεί στην οθόνη κατά τις παρακάτω κλήσεις;

1) ΚΑΛΕΣΕ Πίνακας(A, $A[2]$)

2) ΚΑΛΕΣΕ Πίνακας(A, $A[4]$)

26. Έστω το παρακάτω πρόγραμμα που διαβάζει τα 100 στοιχεία ενός πίνακα και στη συνέχεια υπολογίζει το ελάχιστο μεταξύ των N πρώτων στοιχείων του, χρησιμοποιώντας μια συνάρτηση με όνομα Ελάχιστο:

ΠΡΟΓΡΑΜΜΑ ΜικρότερηΤιμή

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, N

ΠΡΑΓΜΑΤΙΚΕΣ: $A[100], \mu\kappa$

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΔΙΑΒΑΣΕ $A[i]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ N

$\mu\kappa \leftarrow \text{Ελάχιστο}(A, N)$

ΓΡΑΨΕ $\mu\kappa$

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Τέσσερεις μαθητές έγραψαν τις παρακάτω συναρτήσεις Ελάχιστο. Να τις σχολιάσετε ως προς την ορθότητά τους.

ΣΥΝΑΡΤΗΣΗ Ελάχιστο(A, N): ΠΡΑΓΜΑΤΙΚΗ ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: i, N ΠΡΑΓΜΑΤΙΚΕΣ: $A[100], \mu\kappa$ ΑΡΧΗ $\mu\kappa \leftarrow A[1]$ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΑΝ $A[i] < \mu\kappa$ ΤΟΤΕ $\mu\kappa \leftarrow A[i]$ ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ	ΣΥΝΑΡΤΗΣΗ Ελάχιστο(A, N): ΠΡΑΓΜΑΤΙΚΗ ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: i, N ΠΡΑΓΜΑΤΙΚΕΣ: $A[100], \min$ ΑΡΧΗ $\min \leftarrow A[1]$ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΑΝ $A[i] < \min$ ΤΟΤΕ $\min \leftarrow A[i]$ ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ $\text{Ελάχιστο} \leftarrow \min$ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
ΣΥΝΑΡΤΗΣΗ Ελάχιστο($\Pi\text{ιν}, A$): ΠΡΑΓΜΑΤΙΚΗ ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: A, i ΠΡΑΓΜΑΤΙΚΕΣ: $\Pi\text{ιν}[100], \mu\kappa$ ΑΡΧΗ	ΣΥΝΑΡΤΗΣΗ Ελάχιστο(N, A): ΠΡΑΓΜΑΤΙΚΗ ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: $A[100], i, N$ ΠΡΑΓΜΑΤΙΚΕΣ: $\mu\kappa$ ΑΡΧΗ

μικ←Πιν[1] ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ A ΑΝ Πιν[i]<μικ ΤΟΤΕ μικ←Πιν[i] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ Ελάχιστο←μικ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ	μικ←A[1] ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΑΝ A[i]<μικ ΤΟΤΕ μικ←A[i] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ Ελάχιστο←μικ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
--	--

27. Να υλοποιήσετε υποπρόγραμμα που θα δέχεται έναν πίνακα A 10 x 5 και θα επιστρέφει έναν άλλο πίνακα B 10 στοιχείων, που το κάθε του στοιχείο θα είναι το άθροισμα των στοιχείων της αντίστοιχης γραμμής του πίνακα A.

28. Σε ένα διαγωνισμό σφαιροβολίας συμμετέχουν 120 αθλητές. Ο κάθε αθλητής εκτελεί 10 βολές και η τελική του βαθμολογία είναι ο μέσος όρος των 3 καλύτερων βολών του.

i. Να υλοποιήσετε υποπρόγραμμα το οποίο θα διαβάζει το όνομα κάποιου αθλητή και στη συνέχεια διαδοχικά τους βαθμούς των 10 βολών του.

ii. Να αναπτύξετε συνάρτηση που θα δέχεται ως είσοδο έναν πίνακα 10 στοιχείων με τις βολές κάποιου αθλητή και θα επιστρέφει την τελική του βαθμολογία.

iii. Να αναπτύξετε διαδικασία η οποία θα βρίσκει τη θέση του μεγαλύτερου αριθμού που υπάρχει σε έναν μονοδιάστατο πίνακα 120 στοιχείων.

iv. Να χρησιμοποιήσετε τα παραπάνω υποπρογράμματα για να φτιάξετε κύριο πρόγραμμα το οποίο θα εμφανίζει το όνομα του νικητή.

29. Σε ένα πρωτάθλημα μπάσκετ συμμετέχουν 14 ομάδες που η καθεμιά δίνει 26 παιχνίδια. Να γράψετε πρόγραμμα το οποίο:

i. Θα διαβάζει το όνομα και τους πόντους που πέτυχε κάθε ομάδα σε κάθε αγώνα.

ii. Θα διαβάζει ένα όνομα ομάδας και αν υπάρχει θα εμφανίζει τους πόντους που πέτυχε σε κάθε αγώνα.

iii. Θα εμφανίζει τις ομάδες ξεκινώντας από αυτή που πέτυχε τους περισσότερους πόντους και καταλήγοντας σε αυτή που πέτυχε τους λιγότερους σε ολόκληρο το πρωτάθλημα.

Για το ερώτημα (ii) να γράψετε διαδικασία η οποία θα δέχεται έναν πίνακα 14 θέσεων τύπου χαρακτήρα, θα διαβάζει ένα όνομα και θα επιστρέφει τη θέση που βρίσκεται το όνομα αυτό στον πίνακα. Αν το όνομα δεν υπάρχει θα επιστρέφει την τιμή 0. Για το ερώτημα (iii) να γράψετε διαδικασία η οποία θα δέχεται δύο παράλληλους μονοδιάστατους πίνακες, έναν τύπου χαρακτήρα και έναν τύπου ακεραίου 14 στοιχείων και θα τους ταξινομεί σε φθίνουσα σειρά ως προς τον πίνακα τύπου ακεραίου.

ΜΕΤΑΤΡΟΠΗ ΤΜΗΜΑΤΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΙΣΟΔΥΝΑΜΟ ΥΠΟΠΡΟΓΡΑΜΜΑ

ΛΥΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΑ

30. Δίνεται το παρακάτω πρόγραμμα. Να ξαναγραφεί ώστε τα τμήματα A και B να αποτελούν συναρτήσεις αν αυτό είναι δυνατόν, αλλιώς διαδικασίες. (θεωρήστε ότι $\beta \neq 0$).

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: γ, δ

ΑΚΕΡΑΙΕΣ: α, β, temp

ΑΡΧΗ

ΔΙΑΒΑΣΕ α, β

ΑΝ $\alpha > \beta$ **ΤΟΤΕ**

temp ← α

α ← β

β ← temp

γ ← α/β

ΤΕΛΟΣ_ΑΝ

A

ΓΡΑΨΕ α, β

α ← $A_M(\alpha + \beta / 2)$

δ ← $0.4 * \alpha + 0.6 * \beta - \gamma$

B

ΓΡΑΨΕ δ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΛΥΣΗ

Το τμήμα Α δέχεται δύο ακέραιες μεταβλητές α, β και αν το α είναι μεγαλύτερο του β , τις αντιμεταθέτει και στη μεταβλητή γ βάζει το ηλίκο α/β . Επειδή λοιπόν αλλάζει τα περιεχόμενα των μεταβλητών α και β και επειδή αυτές χρησιμοποιούνται παρακάτω στο πρόγραμμα, δεν μπορούμε να χρησιμοποιήσουμε συνάρτηση (η συνάρτηση δεν μεταβάλλει τις παραμέτρους της). Στη διαδικασία που θα φτιάξουμε πρέπει να είμαστε προσεκτικοί στη δήλωση του τύπου των μεταβλητών ώστε να συμφωνούν με αυτές του κυρίως προγράμματος:

ΔΙΑΔΙΚΑΣΙΑ ΔιαδΑ(α, β, γ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α, β, τ

ΠΡΑΓΜΑΤΙΚΕΣ: γ

ΑΡΧΗ

ΑΝ $\alpha > \beta$ **ΤΟΤΕ**

$\tau \leftarrow \alpha$

$\alpha \leftarrow \beta$

$\beta \leftarrow \tau$

$\gamma \leftarrow \alpha/\beta$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Για το τμήμα Β μπορούμε να χρησιμοποιήσουμε συνάρτηση γιατί δεν χρησιμοποιούνται εντολές εισόδου/εξόδου και επίσης η μοναδική μεταβλητή που χρησιμοποιείται παρακάτω στο κυρίως πρόγραμμα είναι η δ , η οποία και διαμορφώνεται μέσα στο τμήμα Β. Η συνάρτηση θα δέχεται τις τιμές των τριών μεταβλητών α, β, γ και θα επιστρέφει την τιμή της δ :

ΣΥΝΑΡΤΗΣΗ ΣυναρΒ(α, β, γ): **ΠΡΑΓΜΑΤΙΚΗ**

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α, β

ΠΡΑΓΜΑΤΙΚΕΣ: γ

ΑΡΧΗ

$\alpha \leftarrow A_M(\alpha + \beta / 2)$

$\text{ΣυναρΒ} \leftarrow 0.4 * \alpha + 0.6 * \beta - \gamma$

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Το κυρίως πρόγραμμα που θα χρησιμοποιεί τα δύο υποπρογράμματα θα είναι:

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: γ

ΑΚΕΡΑΙΕΣ: α, β

ΑΡΧΗ

ΔΙΑΒΑΣΕ α, β

ΚΑΛΕΣΕ ΔιαδΑ(α, β, γ)

ΓΡΑΨΕ α, β

ΓΡΑΨΕ ΣυναρΒ(α, β, γ)

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΑΣΚΗΣΕΙΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

31. Να ξαναγράψετε το παρακάτω πρόγραμμα ώστε τα τμήματα Α, Β και Γ να αποτελούν υποπρογράμματα.

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Ν, βαθμός, ΒΑΘΜΟΙ[30], μικ, θέση, i, Σ

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΑΡΧΗ

$N \leftarrow 30$

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν

ΔΙΑΒΑΣΕ βαθμός

ΟΣΟ βαθμός > 20 **Ή** βαθμός < 1 **ΕΠΑΝΑΛΑΒΕ**

ΓΡΑΨΕ 'Βαθμός εκτός ορίων. Ξαναδώστε βαθμό'

ΔΙΑΒΑΣΕ βαθμός

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Α

$\text{ΒΑΘΜΟΙ}[i] \leftarrow \text{βαθμός}$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

μικ←ΒΑΘΜΟΙ[1]
θέση←1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N
  ΑΝ ΒΑΘΜΟΙ[i]<μικ ΤΟΤΕ
    μικ←ΒΑΘΜΟΙ[i]
    θέση←i
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ θέση

```

B

```

Σ←0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
  Σ←Σ+ΒΑΘΜΟΙ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ←Σ/Ν

```

```

ΑΝ ΜΟ<10 ΤΟΤΕ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
    ΑΝ ΒΑΘΜΟΙ[i]<20 ΤΟΤΕ
      ΒΑΘΜΟΙ[i]←ΒΑΘΜΟΙ[i]+1
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ

```

Γ

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
  ΓΡΑΨΕ ΒΑΘΜΟΙ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

32. Το πρόγραμμα για τον υπολογισμό της παράστασης:

$$A = 1 + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \dots + \frac{1}{N!}x^N$$

είναι το εξής:

ΠΡΟΓΡΑΜΜΑ Παράσταση
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: N,i,P,j
ΠΡΑΓΜΑΤΙΚΕΣ: x,A
ΑΡΧΗ

ΔΙΑΒΑΣΕ x,N

```

A←1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N

```

```

  P←1
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ i
    P←P * j
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

A←A + xⁱ/P

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το αποτέλεσμα είναι:',A

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Το πλαίσιο A αποτελεί το τμήμα προγράμματος που υπολογίζει το i! (i παραγοντικό) , ενώ το πλαίσιο B χρησιμοποιεί το παραγοντικό ώστε να υλοποιήσει την παράσταση A.

i. Να αναπτύξετε το τμήμα A του προγράμματος σε συνάρτηση.

ii. Να αναπτύξετε το τμήμα B του προγράμματος ώστε να χρησιμοποιεί τη συνάρτηση του προηγούμενου ερωτήματος.

iii. Να ξαναγράψετε όλο το πρόγραμμα ώστε να χρησιμοποιεί τη συνάρτηση του ερωτήματος ii.

33. Έστω το παρακάτω πρόγραμμα:

ΠΡΟΓΡΑΜΜΑ Άσκηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100,50],γρ,στ,Σ[100],max,θ_max

ΑΡΧΗ

```

ΓΙΑ γρ ΑΠΟ 1 ΜΕΧΡΙ 100
  ΓΙΑ στ ΑΠΟ 1 ΜΕΧΡΙ 50
    ΔΙΑΒΑΣΕ A[γρ,στ]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

A

ΓΙΑ γρ ΑΠΟ 1 ΜΕΧΡΙ 100

```
Σ[γρ]←0
ΓΙΑ στ ΑΠΟ 1 ΜΕΧΡΙ 50
  Σ[γρ]←Σ[γρ]+Α[γρ,στ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```
max←Σ[1]
θ_max←1
ΓΙΑ γρ ΑΠΟ 2 ΜΕΧΡΙ 100
  ΑΝ Σ[γρ]>max ΤΟΤΕ
    max←Σ[γρ]
    θ_max←γρ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ

ΓΡΑΨΕ θ_max

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

- Να γράψετε διαδικασία που υλοποιεί το τμήμα Α.
- Να γράψετε διαδικασία που υλοποιεί το τμήμα Β.
- Να γράψετε συνάρτηση που υλοποιεί το τμήμα Γ.
- Να γράψετε ισοδύναμο πρόγραμμα που χρησιμοποιεί τα προηγούμενα υποπρογράμματα.

ΛΥΜΕΝΑ ΘΕΜΑΤΑ ΕΞΕΤΑΣΕΩΝ

34. Σε ένα πάρκινγκ η χρέωση γίνεται κλιμακωτά όπως φαίνεται στον παρακάτω πίνακα:

ΔΙΑΡΚΕΙΑ ΣΤΑΘΜΕΥΣΗΣ	ΚΟΣΤΟΣ ΑΝΑ ΩΡΑ
Μέχρι και 3 ώρες	2 ευρώ
Πάνω από 3 και έως 5 ώρες	1,5 ευρώ
Πάνω από 5 ώρες	1,3 ευρώ

I. Να κατασκευαστεί πρόγραμμα το οποίο:

α. περιλαμβάνει τμήμα δηλώσεων.

Μονάδες 2

β.i. διαβάζει τον αριθμό κυκλοφορίας μέχρι να δοθεί το 0. Να θεωρήσετε ότι ο αριθμός κυκλοφορίας μπορεί να περιέχει τόσο γράμματα όσο και αριθμούς.

Μονάδες 2

ii. διαβάζει τη διάρκεια στάθμευσης σε ώρες και τη δέχεται μόνο εφόσον είναι μεγαλύτερη του 0.

Μονάδες 3

iii. καλεί υποπρόγραμμα για τον υπολογισμό του ποσού που πρέπει να πληρώσει ο κάτοχός του.

Μονάδες 2

iv. εμφανίζει τον αριθμό κυκλοφορίας και το ποσό που του αναλογεί.

Μονάδες 2

γ. εμφανίζει το πλήθος των αυτοκινήτων που έμειναν στο πάρκινγκ μέχρι και 2 ώρες.

Μονάδες 4

II. Να κατασκευάσετε το υποπρόγραμμα που καλείται στο ερώτημα β(iii).

Μονάδες 5

ΛΥΣΗ

ΠΡΟΓΡΑΜΜΑ ΠΑΡΚΙΝΓΚ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ΠΛΗΘΟΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ΩΡΕΣ, ΠΟΣΟ

ΧΑΡΑΚΤΗΡΕΣ: ΑΚ

ΑΡΧΗ

ΠΛΗΘΟΣ ← 0

ΔΙΑΒΑΣΕ ΑΚ

ΟΣΟ ΑΚ <> '0' ΕΠΑΝΑΛΑΒΕ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ΩΡΕΣ

ΜΕΧΡΙΣ_ΟΤΟΥ ΩΡΕΣ>0

ΠΟΣΟ ← ΥΠΟΛΟΓΙΣΜΟΣ(ΩΡΕΣ)

ΓΡΑΨΕ ΑΚ , ΠΟΣΟ

ΑΝ ΩΡΕΣ <= 2 ΤΟΤΕ

ΠΛΗΘΟΣ ← ΠΛΗΘΟΣ + 1

ΤΕΛΟΣ_ΑΝ

ΔΙΑΒΑΣΕ ΑΚ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ ΥΠΟΛΟΓΙΣΜΟΣ (X): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ : X, ΧΡΕΩΣΗ

ΑΡΧΗ

ΑΝ $X \leq 3$ ΤΟΤΕ

$ΧΡΕΩΣΗ \leftarrow 2 * X$

ΑΛΛΙΩΣ_ΑΝ $X \leq 5$ ΤΟΤΕ

$ΧΡΕΩΣΗ \leftarrow 3 * 2 + (X - 3) * 1,5$

ΑΛΛΙΩΣ

$ΧΡΕΩΣΗ \leftarrow 3 * 2 + 2 * 1,5 + (X - 5) * 1,3$

ΤΕΛΟΣ_ΑΝ

$ΥΠΟΛΟΓΙΣΜΟΣ \leftarrow ΧΡΕΩΣΗ$

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

35. Μια σύγχρονη πτηνοτροφική μονάδα παρακολουθεί την ημερήσια παραγωγή αυγών και καταγράφει τα στοιχεία σε ηλεκτρονικό αρχείο. Να αναπτύξετε αλγόριθμο ο οποίος θα διαχειρίζεται τα στοιχεία της μονάδας στη διάρκεια ενός έτους. Για το σκοπό αυτό:

A. Να κατασκευάσετε κύριο πρόγραμμα το οποίο:

1. να ζητάει το έτος παρακολούθησης, ελέγχοντας ότι πρόκειται για έτος του 21^{ου} αιώνα (από 2000 μέχρι και 2099). Ο αλγόριθμος να δημιουργεί πίνακα με τον αριθμό των ημερών για καθέναν από τους δώδεκα μήνες του έτους που δόθηκε. Ο αριθμός των ημερών του μήνα θα υπολογίζεται από υποπρόγραμμα το οποίο θα κατασκευάσετε για το σκοπό αυτό. Η λειτουργία του υποπρογράμματος περιγράφεται στο ερώτημα B.

Μονάδες 3

2. να ζητάει την ημερήσια παραγωγή (αριθμό αυγών) για κάθε μέρα του έτους και να καταχωρίζει τις τιμές σε πίνακα δύο διαστάσεων, με μια γραμμή για κάθε μήνα.

Μονάδες 3

3. να εμφανίζει τον τρίτο κατά σειρά από τους μήνες του έτους που έχουν ο καθένας μέσο όρο ημερήσιας παραγωγής μέχρι και δέκα ποσοστιαίες μονάδες πάνω ή κάτω από τον ετήσιο μέσο όρο. Αν δεν βρει τέτοιο μήνα, να εμφανίζει κατάλληλο μήνυμα.

Μονάδες 8

B. Να κατασκευάσετε υποπρόγραμμα το οποίο να δέχεται ως παραμέτρους κάποιο έτος και τον αριθμό κάποιου μήνα (1 έως 12), και να επιστρέφει τον αριθμό των ημερών του συγκεκριμένου μήνα. Όταν το έτος είναι δίσεκτο, ο Φεβρουάριος έχει 29 ημέρες, διαφορετικά έχει 28. Δίσεκτα είναι τα έτη που διαιρούνται με το 4 αλλά όχι με το 100, καθώς και εκείνα που διαιρούνται με το 400. Για τους υπόλοιπους μήνες, πλην του Φεβρουαρίου, ισχύει το εξής: μέχρι και τον Ιούλιο (7ος μήνας) οι μονοί μήνες έχουν 31 ημέρες και οι ζυγοί 30. Για τους μήνες μετά τον Ιούλιο, ισχύει το αντίστροφο.

Μονάδες 6

ΛΥΣΗ

Θα χρησιμοποιήσουμε έναν διδιάστατο πίνακα 12 X 31 (12 μήνες X 31 ημέρες μέγιστο που μπορεί να έχει κάθε μήνας) για να αποθηκεύσουμε την ημερήσια παραγωγή αυγών για κάθε μέρα του έτους. Για τους μήνες του έτους που δεν έχουν 31 ημέρες, η αντίστοιχη γραμμή του πίνακα θα περιέχει τιμές μέχρι την τελευταία ημέρα του μήνα π.χ. για τον 4^ο μήνα (Απρίλιο) που έχει 30 ημέρες, η 4^η γραμμή του πίνακα θα περιέχει 30 τιμές από τις 31 θέσεις της γραμμής του πίνακα. Αντίστοιχα, για το 2^ο μήνα (Φεβρουάριο), η 2^η γραμμή του πίνακα θα περιέχει 28 ή 29 τιμές ανάλογα με το αν το έτος είναι δίσεκτο ή όχι (στα δίσεκτα έτη ο Φεβρουάριος έχει 29 ημέρες). Στη συνέχεια, θα φτιάξουμε τον πίνακα ΜΟ 12 θέσεων που θα περιέχει το μέσο όρο ημερήσιας παραγωγής για κάθε μήνα του έτους και το πίνακα ΠΔ 12 θέσεων που θα περιέχει για κάθε μήνα την ποσοστιαία διαφορά του κάθε μήνα από τον ετήσιο μέσο όρο. Οι πίνακες θα είναι οι εξής:

HM_Π	1	2	...	31		ΜΟ		ΠΔ
1					→	1		1
2					→	2		2
3					→	3		3
.....					→			
12					→	12		12

Για το ερώτημα 3 θα πρέπει να κάνουμε σειριακή αναζήτηση στον πίνακα ΠΔ και όταν βρίσκουμε στοιχείο του πίνακα μικρότερο ή ίσο με 10, να αυξάνουμε έναν μετρητή κατά ένα. Όταν ο μετρητής γίνει 3, θα έχουμε βρει το μήνα που ψάχνουμε.

ΠΡΟΓΡΑΜΜΑ ΠΤΗΝΟΤΡΟΦΟΣ ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : Έτος,Σύνολο_Έτους,Π,Ι,Ι,Σύνολο_μήνα,Ημ_Μήνα, ΗΜ_Π[12, 31],Κ,ΜΗΝΑΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ_ΕΤΟΥΣ, ΜΟ[12],ΠΔ[12]

ΛΟΓΙΚΕΣ: Βρέθηκε

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Έτος

ΜΕΧΡΙΣ_ΟΤΟΥ Έτος >=2000 **ΚΑΙ** Έτος <=2099

} διάβασμα του έτους με έλεγχο εγκυρότητας

Σύνολο_έτους ← 0

!μηδενισμός του συνόλου ημερήσιας παραγωγής όλου του έτους

Π ← 0

!μηδενισμός του μετρητή ημερών του έτους

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 12

Σύνολο_μήνα ← 0

!μηδενισμός του συνόλου ημερήσιας παραγωγής για τον Ι μήνα

Ημ_μήνα ← Αρ_Ημερών(Έτος, Ι) !υπολογισμός των ημερών του Ι μήνα με κλήση της συνάρτησης

ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Ημ_μήνα

ΔΙΑΒΑΣΕ ΗΜ_Π[I, J]

Σύνολο_μήνα ← Σύνολο_μήνα + ΗΜ_Π[I, J]

Σύνολο_έτους ← Σύνολο_έτους + ΗΜ_Π[I, J]

Π ← Π + 1

} διάβασμα του πίνακα ΗΜ_Π και
ταυτόχρονα υπολογισμός του συνόλου
ετήσιας παραγωγής, μηνιαίας παραγωγής
και συνόλου ημερών του έτους

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ[I] ← Σύνολο_μήνα / Ημ_μήνα !υπολογισμός του πίνακα ΜΟ με τους μηνιαίους μέσους όρους

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ_ΕΤΟΥΣ ← Σύνολο_έτους / Π

!υπολογισμός του ετήσιου μέσου όρου

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 12

ΠΔ[I] ← (Α_Τ(ΜΟ_ΕΤΟΥΣ - ΜΟ[I]) / ΜΟ_ΕΤΟΥΣ) *100

} υπολογισμός του πίνακα ΠΔ με την ποσοστιαία
διαφορά του μέσου όρου κάθε μήνα από τον
ετήσιο μέσο όρο

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Κ ← 0 !μηδενισμός του μετρητή των μηνών που έχουν ποσοστιαία διαφορά μικρότερη ή ίση με 10

Ι ← 1

Βρέθηκε ← ψευδής

ΟΣΟ Ι <=12 **ΚΑΙ** Βρέθηκε = ψευδής **ΕΠΑΝΑΛΑΒΕ**

ΑΝ ΠΔ[I] <= 10 **ΤΟΤΕ**

Κ ← Κ +1

ΤΕΛΟΣ_ΑΝ

ΑΝ Κ = 3 **ΤΟΤΕ**

Βρέθηκε ← αληθής

ΜΗΝΑΣ ← Ι

ΑΛΛΙΩΣ

Ι ← Ι + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ Βρέθηκε = αληθής **ΤΟΤΕ**

ΓΡΑΨΕ ΜΗΝΑΣ

ΑΛΛΙΩΣ

ΓΡΑΨΕ ' ΔΕΝ ΥΠΑΡΧΕΙ ΑΝΤΙΣΤΟΙΧΟΣ ΜΗΝΑΣ '

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Αρ_Ημερών(Έτος, Μήνας): **ΑΚΕΡΑΙΑ**

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : Έτος, Μήνας

ΑΡΧΗ

ΑΝ Μήνας <> 2 **ΤΟΤΕ**

ΑΝ (Μήνας<=7 **ΚΑΙ** Μήνας MOD 2=0) **Ή** (Μήνας>7 **ΚΑΙ** Μήνας MOD 2=1) **ΤΟΤΕ**

Αρ_Ημερών ← 30

ΑΛΛΙΩΣ

Αρ_Ημερών ← 31

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

ΑΝ (Έτος MOD 4=0 **ΚΑΙ** Έτος MOD 100<>0) **Ή** (Έτος MOD 400=0) **ΤΟΤΕ**

Αρ_Ημερών ← 29

ΑΛΛΙΩΣ

Αρ_Ημερών ← 28

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ

36. Μια εταιρεία ενοικίασης αυτοκινήτων έχει νοικιάσει 30 αυτοκίνητα, τα οποία κατηγοριοποιούνται σε οικολογικά και συμβατικά. Η πολιτική χρέωσης για την ενοικίαση ανά κατηγορία και ανά ημέρα δίνεται στον παρακάτω πίνακα.

ΗΜΕΡΕΣ	ΟΙΚΟΛΟΓΙΚΑ	ΣΥΜΒΑΤΙΚΑ
1-7	30 € ανά ημέρα	40 € ανά ημέρα
8-16	20 € ανά ημέρα	30 € ανά ημέρα
Από 17 και άνω	10 € ανά ημέρα	20 € ανά ημέρα

Να αναπτύξετε πρόγραμμα το οποίο:

α. περιλαμβάνει τμήμα δηλώσεων μεταβλητών.

β. για κάθε αυτοκίνητο που έχει ενοικιαστεί:

i) διαβάσει την κατηγορία του («ΟΙΚΟΛΟΓΙΚΑ» ή «ΣΥΜΒΑΤΙΚΑ») και τις ημέρες ενοικίασης.

ii) καλεί υποπρόγραμμα με είσοδο την κατηγορία του αυτοκινήτου και τις ημέρες ενοικίασης και υπολογίζει με βάση τον παραπάνω πίνακα τη χρέωση.

iii) εμφανίζει το μήνυμα 'χρέωση' και τη χρέωση που υπολογίσατε.

γ. υπολογίζει και εμφανίζει το πλήθος των οικολογικών και συμβατικών αυτοκινήτων.

δ. Να κατασκευάσετε το κατάλληλο υποπρόγραμμα του ερωτήματος (β) (ii).

Σημείωση:

1. Δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου.

2. Ο υπολογισμός της χρέωσης δεν πρέπει να γίνει κλιμακωτά.

(Εξετάσεις 2008)

37. Στην εταιρεία διοδίων χρησιμοποιούν πίνακα Π[3] που περιέχει το πλήθος των οχημάτων που έχουν διέλθει από τα διόδια, πιο συγκεκριμένα: Επιβατικά (θέση 1 πίνακα), Φορτηγά (θέση 2 πίνακα) και Δίκυκλα (θέση 3 πίνακα). Το κόστος διέλευσης για κάθε κατηγορία οχήματος είναι 2, 3, 1 ευρώ με τη σειρά που αναφέρθηκαν. Να αναπτύξετε αλγόριθμο ή πρόγραμμα για την ημερήσια διαχείριση των διοδίων που θα επιτελεί τις παρακάτω λειτουργίες:

α. Θα μηδενίζει τα κελιά του πίνακα Π (αρχή της ημέρας).

β. Θα δέχεται επαναληπτικά τον τύπο ενός διερχόμενου οχήματος. Να θεωρήσετε γνωστή τη διαδικασία (δε χρειάζεται να την υλοποιήσετε) «Διέλευση_οχήματος» που επιστρέφει τον τύπο του οχήματος, που μπορεί να είναι "Ε", "Φ" ή "Δ".

γ. Να εμφανίζει το κόστος διέλευσης του οχήματος αυτού. Ο υπολογισμός του κόστους θα πραγματοποιείται με υποπρόγραμμα που θα υλοποιηθεί για το σκοπό αυτό.

δ. Θα ενημερώνει τον πίνακα Π καταμετρώντας το διερχόμενο όχημα. Η ενέργεια αυτή θα πραγματοποιείται με υποπρόγραμμα που θα υλοποιηθεί για το σκοπό αυτό. Η επαναληπτική διαδικασία ολοκληρώνεται όταν εισαχθεί το "ΤΕΛΟΣ" ως τύπος οχήματος.

ε. Θα εμφανίζεται το πλήθος των οχημάτων κάθε κατηγορίας που διήλθαν. Η ενέργεια αυτή θα πραγματοποιείται με υποπρόγραμμα που θα υλοποιηθεί για το σκοπό αυτό.

38. Να αναπτυχθούν τα υποπρογράμματα και το πρόγραμμα που αναφέρονται στα επόμενα ερωτήματα.

α. Να γίνει διαδικασία με όνομα «αναζήτηση» η οποία να δέχεται ως παραμέτρους έναν πίνακα χαρακτήρων «ΠΙΝΑΚΑΣ» 10000 στοιχείων, μια ακέραια μεταβλητή Ν και μια μεταβλητή «ζητούμενο» τύπου χαρακτήρα και να επιστρέφει μέσω ενός πίνακα «ΑΠΟΤΕΛΕΣΜΑΤΑ» σε ποιες από τις Ν πρώτες θέσεις του πίνακα «ΠΙΝΑΚΑΣ» υπάρχει το στοιχείο «ζητούμενο» και μέσω μιας μεταβλητής «βρέθηκαν» το πλήθος των θέσεων αυτών.

β. Να γίνει διαδικασία η οποία θα διαβάσει τον αριθμό κυκλοφορίας (για παράδειγμα ΑΧΕ 3421), την μάρκα και το χρώμα το πολύ 10000 αυτοκινήτων και θα τα καταχωρεί σε αντίστοιχους πίνακες τύπου χαρακτήρων με ονόματα «ΑΚ», «ΜΑΡΚΑ», «ΧΡΩΜΑ». Μετά από κάθε καταχώρηση θα ερωτάται ο χρήστης αν επιθυμεί άλλη καταχώρηση. Αν η απάντηση είναι ΝΑΙ θα καταχωρούνται τα στοιχεία του επόμενου αυτοκινήτου, ενώ αν η απάντηση είναι ΟΧΙ, τότε θα διακόπτεται η εκτέλεση της διαδικασίας και θα επιστρέφονται οι πίνακες καθώς και το πλήθος των στοιχείων που καταχωρήθηκαν.

γ. Να γίνει πρόγραμμα που θα καλεί τα παραπάνω υποπρογράμματα και αφού διαβάσει τον αριθμό κυκλοφορίας, την μάρκα και το χρώμα των αυτοκινήτων που εισήλθαν στο κέντρο της Αθήνας θα εκτελεί τις εξής λειτουργίες :

i. Θα διαβάσει έναν αριθμό κυκλοφορίας και στην περίπτωση που το συγκεκριμένο όχημα εισήλθε στο κέντρο της πόλης, θα εμφανίζει το χρώμα και την μάρκα του. Σε διαφορετική περίπτωση θα εμφανίζει το μήνυμα « Το συγκεκριμένο όχημα δεν εισήλθε στο κέντρο».

ii. Θα εμφανίζει τον αριθμό κυκλοφορίας και την μάρκα των οχημάτων που εισήλθαν στο κέντρο της πόλης και το χρώμα τους ήταν μπλε.

39. Να κάνετε πρόγραμμα το οποίο θα δέχεται ως είσοδο την έκταση μιας πόλης σε τετραγωνικά μέτρα, το όνομά της και το πλήθος των κατοίκων της. Πυκνοκατοικημένη θεωρείται μια πόλη όταν φιλοξενεί τουλάχιστον 0.5 κατοίκους ανά τετραγωνικό μέτρο. Το πρόγραμμα θέλουμε να ικανοποιεί τις εξής προδιαγραφές:

i. Να τερματίζει όταν δοθεί ως είσοδος για μια πόλη 0 τετραγωνικά μέτρα έκταση.

ii. Να καλεί υποπρόγραμμα (το οποίο θα δημιουργήσετε) που θα επιστρέφει μια μεταβλητή λογικού τύπου, η οποία θα παίρνει τιμή *αληθής* αν κάποια πόλη είναι πυκνοκατοικημένη. Το κύριο πρόγραμμα θα καλεί το υποπρόγραμμα αυτό για να εξετάζει αν η κάθε πόλη είναι πυκνοκατοικημένη ή όχι. Σε περίπτωση μη πυκνοκατοικημένης πόλης θα εμφανίζει κατάλληλο μήνυμα.

iii. Να μετρά το πλήθος των πυκνοκατοικημένων πόλεων.

40. Μια μέθοδος για την ανίχνευση υψηλών θερμοκρασιών σε κάποια περιοχή είναι ο υπολογισμός του μέσου όρου των υψηλότερων μηνιαίων θερμοκρασιών που παρουσιάστηκαν στη συγκεκριμένη περιοχή κατά τη διάρκεια του έτους. Ακολουθώντας τα παρακάτω στάδια να υλοποιήσετε πρόγραμμα το οποίο θα υπολογίζει τον συγκεκριμένο μέσο όρο.

i. Να αναπτύξετε διαδικασία η οποία θα διαβάζει και θα καταχωρεί σε πίνακα 30x12 τις θερμοκρασίες που παρουσιάστηκαν σε όλο το έτος στη συγκεκριμένη περιοχή. Υποθέτουμε ότι όλοι οι μήνες έχουν 30 ημέρες.

ii. Να αναπτύξετε διαδικασία η οποία θα δέχεται ως είσοδο τον διδιάστατο πίνακα με όλες τις θερμοκρασίες και θα κατασκευάζει έναν μονοδιάστατο πίνακα, 12 στοιχείων, ο οποίος σε κάθε θέση του θα περιέχει τη μέγιστη θερμοκρασία που παρουσιάστηκε τον συγκεκριμένο μήνα. Για το ερώτημα αυτό θεωρήστε δεδομένη και χρησιμοποιήστε τη συνάρτηση ΜΕΓΙΣΤΟ(Α), όπου Α είναι ένας μονοδιάστατος πίνακας 30 στοιχείων, η οποία επιστρέφει το μέγιστο στοιχείο του πίνακα αυτού.

iii. Να υλοποιήσετε κύριο πρόγραμμα το οποίο θα χρησιμοποιεί τα υποπρογράμματα των ερωτημάτων (i) και (ii) και θα εμφανίζει τη μέση τιμή των μέγιστων θερμοκρασιών που εμφανίστηκαν σε κάθε μήνα του χρόνου.

41. Στο άθλημα των 110 μέτρων μετ' εμποδίων, στους δύο ημιτελικούς αγώνες συμμετέχουν δέκα έξι (16) αθλητές (8 σε κάθε ημιτελικό). Σύμφωνα με τον κανονισμό στον τελικό προκρίνεται ο πρώτος αθλητής κάθε ημιτελικού. Η οκτάδα του τελικού συμπληρώνεται με τους αθλητές που έχουν τους έξι (6) καλύτερους χρόνους απ' όλους τους υπόλοιπους συμμετέχοντες. Να θεωρήσετε ότι δεν υπάρχουν αθλητές με ίδιους χρόνους.

1. Να γράψετε πρόγραμμα στη «ΓΛΩΣΣΑ» το οποίο

α. περιλαμβάνει το τμήμα δηλώσεων.

β. καλεί τη διαδικασία ΕΙΣΟΔΟΣ για κάθε ημιτελικό ξεχωριστά. Η διαδικασία διαβάζει το όνομα του αθλητή και τον χρόνο του (με ακρίβεια δεκάτου του δευτερολέπτου).

γ. καλεί τη διαδικασία ΤΑΞΙΝΟΜΗΣΗ για κάθε ημιτελικό ξεχωριστά. Η διαδικασία ταξινομεί τους αθλητές ως προς τον χρόνο τους με αύξουσα σειρά.

δ. δημιουργεί τον πίνακα ΟΝ με τα ονόματα και τον πίνακα ΧΡ με τους αντίστοιχους χρόνους των αθλητών που προκρίθηκαν στον τελικό.

ε. εμφανίζει τα ονόματα και τους χρόνους των αθλητών που θα λάβουν μέρος στον τελικό.

2. Να γράψετε

α. τη διαδικασία ΕΙΣΟΔΟΣ.

β. τη διαδικασία ΤΑΞΙΝΟΜΗΣΗ.

42. Να αναπτύξετε πρόγραμμα το οποίο θα διαβάζει τους βαθμούς (1-20) και τα ονόματα το πολύ 100 μαθητών και στη συνέχεια θα εμφανίζει όλους τους μαθητές καθώς και τους βαθμούς τους από τους καλύτερους μέχρι και τους χειρότερους ως εξής:

Βαθμό 20 έχουν:

Κωστόπουλος Χρήστος

Βαθμό 18 έχουν:

Κωστόπουλος Νίκος

Βασιλάς Στρατής

Αντωνιάδης Πάνος

Βαθμό 17 έχουν:

Μπουρούνης Μιχάλης

.....

Στην περίπτωση που κάποιον συγκεκριμένο βαθμό (από 1 έως και 20) δεν τον έχει κανένας μαθητής, τότε, όπως φαίνεται και στο παράδειγμα, να μην εμφανίζεται καθόλου.

Για το πρόγραμμα αυτό να αναπτύξετε και να χρησιμοποιήσετε τα εξής υποπρογράμματα:

i. Διαδικασία η οποία θα διαβάζει τους βαθμούς (τύπου ακεραίου) καθώς και τα ονόματα το πολύ 100 μαθητών. Η διαδικασία εισαγωγής θα ολοκληρώνεται είτε όταν εισαχθούν 100 μαθητές είτε όταν ο χρήστης

δώσει μη έγκυρο βαθμό. Η διαδικασία θα επιστρέφει τους πίνακες στους οποίους εισήχθησαν τα στοιχεία αυτά.

ii. Συνάρτηση η οποία θα δέχεται ως είσοδο έναν πίνακα 100 στοιχείων, μια θέση του πίνακα και έναν ακέραιο αριθμό και θα επιστρέφει την τιμή ΑΛΗΘΗΣ στην περίπτωση που ο συγκεκριμένος αριθμός περιέχεται στα στοιχεία του πίνακα μέχρι την συγκεκριμένη θέση ή την τιμή ΨΕΥΔΗΣ στην αντίθετη περίπτωση.

iii. Διαδικασία η οποία θα εμφανίζει τα ονόματα των μαθητών που έχουν έναν συγκεκριμένο βαθμό.

43. Δίνεται η διαδικασία Ταξινόμηση:

ΔΙΑΔΙΚΑΣΙΑ Ταξινόμηση(A)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i , j

ΠΡΑΓΜΑΤΙΚΕΣ: A[100], temp

ΛΟΓΙΚΕΣ: ταξινομήθηκε

ΑΡΧΗ

ταξινομήθηκε ← ψευδής

i ← 2

ΟΣΟ ταξινομήθηκε <> αληθής **ΕΠΑΝΑΛΑΒΕ**

ταξινομήθηκε ← αληθής

ΓΙΑ j **ΑΠΟ** 100 **ΜΕΧΡΙ** i **ΜΕ ΒΗΜΑ** -1

ΑΝ A[j-1] > A[j] **ΤΟΤΕ**

temp ← A[j]

A[j] ← A[j-1]

A[j-1] ← A[j]

Τμήμα A

Τμήμα B

ταξινομήθηκε ← ψευδής

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

i ← i+1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Η διαδικασία αυτή υλοποιεί τον αλγόριθμο της φυσαλίδας έτσι, ώστε στην περίπτωση που δεν γίνει καμιά αντιμετάθεση σε κάποιο πέρασμα, πράγμα που σημαίνει ότι ο πίνακας έχει ταξινομηθεί, τότε να διακόπτεται η εκτέλεσή της.

i. Να υλοποιήσετε διαδικασία Αντεμετάθεση(A, θ1, θ2) η οποία θα δέχεται δύο ακέραιους αριθμούς θ1, θ2 και έναν πίνακα A, με 100 στοιχεία, και θα αντιμεταθέτει τα στοιχεία του πίνακα που βρίσκονται στις θέσεις θ1 και θ2.

ii. Να υλοποιήσετε με υποπρόγραμμα το τμήμα B. Το υποπρόγραμμα αυτό να καλεί τη διαδικασία του ερωτήματος (i).

iii. Να γράψετε και πάλι τη διαδικασία Ταξινόμηση έτσι, ώστε να χρησιμοποιεί το υποπρόγραμμα που αναπτύξατε στο ερώτημα (ii).